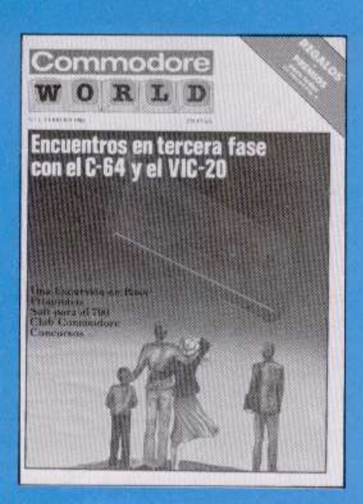
Commodore WORLD

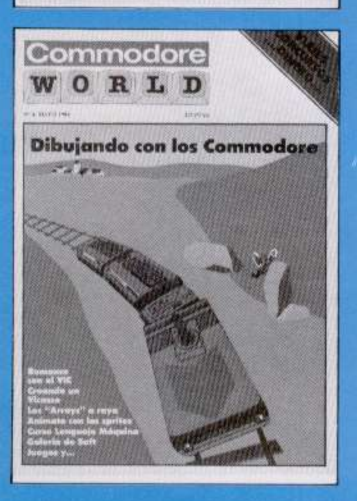
Nº 6 JULIO/AGOSTO 1984 315 PTAS.




Commodore
WORLD
Encuentros en tercera fase con el C-64 y el VIC-20


Commodore
WORLD
El "duende" de los Commodore


Commodore
WORLD
Excursión por la Microinformática


Commodore
WORLD
Dibujando con los Commodore

Commodore WORLD

Commodore World
está publicado por
SIMSA
en colaboración con
MEC-COMMODORE

EQUIPO
Manuel AMADO; Nieves CHESA;
José Luis ERRAZQUIN;
M. A. HERMOSELL;
María LOPEZ;
Juan MARTINEZ;
Pere MASATS; Victoria MORALES;
Rafael PARDO; Diego ROMERO;
Albert SANGLAS; Jordi SASTRE;
Valerie SHANKS...
...Y NUESTROS LECTORES

SIMSA
Coordinador María López
c/ Pedro Muguruza, 4-8ºB. Madrid-16
Tel.: (91) 259 54 78
Delegación en Barcelona:
Sant Gervasi de Cassoles, 39
despacho 4 - Barcelona-22.
Tel.: (93) 212 73 45/212 88 48

MEC-COMMODORE
Coordinador Pere Masats
Taquígrafo Serra, 7-5º
Barcelona-29
Tels.: (93) 250 51 03/02



Imprime:
IBERDOS, S. A.
Germán Pérez Carrasco, 24.
Madrid-27
Depósito Legal: M-2944-1984

SUMARIO

CONTENIDO — **PAG.**

PROXIMO NUMERO

- MANUAL DEL MONITOR DE CODIGO MAQUINA.
- ES LA HORA DE APRENDER INGLES (Dos programas para los peques —o no tanto— en el "cole" o en casa)
- COMO RESUCITAR LOS FICHEROS "ASESINADOS"
- NUMEROS ALEATORIOS... SI, PERO A LO DIVERTIDO.
- JUEGOS
- CLUB COMMODORE
- ... Y TODAS VUESTRAS COLABORACIONES.

¡OS ESPERAMOS!

SUPER-INTERESANTISIMO

CONCURSOS "A GOGO"

Aparte de nuestros concursos permanentes entre los lectores colaboradores (ver pie de página) nacen hoy tres nuevos grandes concursos y nuevos e interesantes alicientes para nuestros colaboradores de programas.

1 Concurso de programación "Ferre Moret" —550.000 en premios—

La casa Ferre-Moret convoca hoy, en colaboración con, y a través de Commodore World, un importante concurso de programación sobre tarjetas de 40-80 columnas para el VIC y de 80 columnas para el C-64 con premios de 50.000 a 250.000 ptas y opción de compra para programas no ganadores. Bases en pág. 6.

2 A por el suscriptor 8.000, un viaje y una impresora.

En mi última visita a Carlos Domenech, presidente de MEC, le comentaba que me encontraba en un anticlimax después de toda la emoción de la carrera hacia el 5.000 y los viajes a Estados Unidos e Inglaterra, "bueno, me dijo riéndose, Commodore tiene instalaciones en Hong-Kong"... Vale, vale, vale... precioso... quizás un día ...¿quién sabe?... pero las finanzas de Commodore World no pueden permitirse ese lujo de momento... pero sí otros lujos más modestos. Cuando lleguemos al número 8.000 se rifará un viaje para dos personas ENTRE TODOS LOS SUSCRIPTORES y el ganador podrá elegir el lugar entre varias opciones (ver pie de página). Asimismo se sorteará ENTRE TODOS LOS SUSCRIPTORES una impresora MPS 801 (o similar) regalo de MEC. Desde aquí quiero dar las gracias a Carlos Domenech por su colaboración.

3 Primer concurso nacional escolar

Ya anunciado en nuestro número anterior y cuyas bases se repiten de nuevo en pág. 6. Se está trabajando a fondo y se presentará oficialmente a la prensa en el mes de septiembre. En nuestro próximo número anunciaremos lugar, día y hora.

4 20% de la venta de cintas para los lectores colaboradores

Inicialmente no pensábamos comercializar los programas remitidos por los lectores y publicados en Commodore World. Sin embargo, últimamente hemos tenido varias peticiones de cintas de programas de colaboradores, por lo que decidimos y ya lo estamos haciendo, remitir un 20% por cinta vendida a los autores correspondientes.

Lo que nos lleva a...

...Nuestros lectores los grandes protagonistas

Commodore World nació como una auténtica revista de equipo, un equipo formado por tres colaboradores: la Editorial SIMSA, Club Commodore de MEC y todos los lectores.

Estos últimos están resultando ser el alma de la revista, marcando la pauta de lo que quieren ver y leer, dándonos la clave de las necesidades de la gran familia Commodoriana y colaborando activamente con sus aportaciones personales, siendo estas tantas que casi no damos abasto para leer, comprobar y publicar. Todos los números nos quedan dos o tres programas "en cartera" para el próximo. Si alguno de vosotros no veis vuestro programa, tened paciencia, pues seguro que aparece en el número siguiente. Ocasionalmente, tenemos que devolver algún programa, por defectuoso o no "entrar". Devolvédnoslo corregido, eso es todo...

...Por eso...

...Estamos ideando siempre nuevos premios y alicientes pues creemos que aparte de divertirnos todos, el tema es importante. La informática no es ya ni un lujo, ni un juguete extravagante, sino una parte más de la forma de vida de este final de siglo, y en un futuro cercanísimo un arma de trabajo casi imprescindible. No queremos ser una revista más de simple entretenimiento, sino también una herramienta de utilidad y de creación de buena informática y plataforma de buenos informáticos (¡presentes y futuros!).

Manuales en castellano de cartuchos MEC

MEC ha tenido una gran idea, el entregarnos para su publicación los manuales en castellano de algunos de sus cartuchos especiales, a fin de que todos los usuarios tengan acceso a los mismos. En este número comenzamos con el del Superexpander para el VIC que publicaremos en dos partes y en el próximo número comenzaremos con el MANUAL DEL MONITOR DE CODIGO MAQUINA, que publicaremos probablemente en tres partes. Otros seguirán que iremos anunciando

Feliz verano a todos

Muchas otras son las cosas que aparecen en este número aunque prefiero que ellas hablen por sí mismas y despedirnos hasta el 1º de septiembre deseando unas felices vacaciones a todos, vosotros y nosotros.

CONCURSOS PERMANENTES

Colaboraciones de Programas y Magia

La verdad sea dicha: os estáis portando; las colaboraciones que nos están llegando son numerosas y de muy buena calidad. Por lo tanto, vamos a aclarar las condiciones de nuestros concursos permanentes.

1) Los sorteos se realizarán en los meses de junio y diciembre.

2) Se sortearán siempre seis premios, tres de 15.000, 10.000 y 5.000 pesetas en metálico de "Commodore World" y tres del mismo valor en material de Microelectrónica y Control, entre todas las colaboraciones publicadas.

3) Se sortearán siempre seis premios, tres de 3.000, 2.000 y 1.000 pesetas en metálico de "Commodore World" y tres del mismo valor en material de Microelectrónica y Control, entre todas las contribuciones publicadas en la sección de "Magia".

4) Adicionalmente, se sortearán diversos premios de cintas, juegos, etc. donados por firmas y distribuciones, entre todas las colaboraciones.

5) Los autores de las colaboraciones vendidas dentro de nuestro "Servicio de Cintas" percibirán el 20% del precio de la cinta.

6) Todas las colaboraciones deben venir escritas a máquina y los programas grabados en cinta (si es posible), o con el listado completo en impresora. Nuestros lectores más jóvenes pueden escribir a mano pero con letra muy clara.

7) Quedan automáticamente descalificadas tanto del sorteo como del "Servicio de Cintas" las colaboraciones que hayan sido enviadas a otras revistas.

8) Las colaboraciones se enviarán a "Commodore World", *C/Pedro Muguruza, 4, 8º B. Madrid-16.*

MAS VIAJES Y OTRO REGALO DE "IMPRESION"... A POR EL SUSCRIPTOR 8.000

Es evidente que a nuestros sucriptores les gusta viajar. De momento Joan Pujiula visitará los Estados Unidos en julio y Juan Mejuto Iglesias irá a Inglaterra en el otoño. Queríamos sortear algo un poco especial ENTRE TODOS LOS SUCRIPTORES cuando lleguemos al suscriptor número 8.000, y hemos optado por dos premios, uno concedido por Microelectrónica y Control y otro concedido por "Commodore World".

MEC regalará una impresora MPS 801 (o similar), y nosotros hemos escogido un surtido de viajes muy "chulos" para dos personas para que el ganador elija el que más le apetezca. La selección que hemos hecho es la siguiente:

Viajes de 4 días en hoteles de primera clase a uno de estos lugares: París, Lisboa, Atenas, Amsterdam, Niza, Viena.

O viajes de una semana en hoteles de 4/5 estrellas: Canarias o Mallorca.





Tres momentos del sorteo que se celebró el día 18 para la entrega de los premios abajo detallado.

GANADORES DEL GRAN SORTEO COMMODORE WORLD

El día 18 de junio se celebró el 1er. sorteo semestral de Commodore World entre todos los colaboradores, así como el del viaje a Inglaterra entre todos los suscriptores. El sorteo se celebró públicamente en el English Montessori School de Madrid.

Queremos expresar nuestro más sincero agradecimiento a:

Milagros King, Directora del English Montessori, que no solamente nos cedió el local sino que además nos preparó una fiesta de Coca Colas-chips y panchitos.

A la casa Indescomp que sólo por su continua dedicación a la promoción de la informática ha hecho posible que pudiésemos entregar 10 premios extras.

A Microinformática y Soft que, a pesar de su reciente creación, ha contribuido con otros cuatro premios.

En nombre de todo el equipo: Simsa, MEC y lectores, gracias a todos ellos.

PREMIOS

Sorteo viaje a Inglaterra - Nº suscriptor: 3.766. Juan Mejuto Iglesias. Santiago de Compostela (La Coruña)

COLABORACIONES

Premios entregados por Commodore World:

—15.000 ptas. Alfredo Sindin Valero. Madrid. Apareció revista nº 5.
—10.000 ptas. José Vicente López Navarrro. Albacete. Apareció revista nº 4.
—5.000 ptas. Felca. Palma de Mallorca (Baleares). Aparece revista nº 6.

Material entregado por Microelectrónica y Control por valor de:

—15.000 ptas. Rafael Muñoz Alfonso. Barcelona. Aparece revista nº 6.
—10.000 ptas. Guillermo Caamaño Olivares. Granada. Apareció revista nº 4.
—5.000 ptas. Club Alaiz. Pamplona (Navarra). Apareció revista nº 3.

2 regla-reloj-calculadoras entregadas por ComputerWorld/España a:

—Carlos Beldarrain Santos. Pamplona (Navarra). Apareció revista nº 5
—Descalificado.

6 cintas entregadas por INDESCOMP a:

—R. C. Massó. Barcelona. Apareció revista nº 6
—Fco. Sarrias Pedemonte. Barcelona. Aparece revista nº 6
—Pedro Obrador Arta. Palma de Mallorca (Baleares). Aparece revista nº 6
—Domingo Mánguez Gallego. Móstoles (Madrid). Apareció revista nº 5
—Juan Carlos Loren. Leganés (Madrid). Aparece revista nº 6
—Luis Carballo Taboada. Orense. Apareció revista nº 1.

MAGIA

Premios entregados por Commodore World:

—3.000 ptas. José María Rodríguez. Las Arenas (Vizcaya). Apareció revista nº 5.
—2.000 ptas. Rafael García Segura. Málaga. Apareció revista nº 4.
—1.000 ptas. Leandro Olalla. Málaga. Apareció revista nº 5.

Material entregado por Microelectrónica y Control por valor de:

—3.000 ptas. J.M.V. Apareció revista nº 3.
—2.000 ptas. Luis Suárez Antuña. Alcalá de Henares (Madrid). Apareció revista nº 5.
—1.000 ptas. Alfredo Sindin Valero. Madrid. Apareció revista nº 3.

2 regla-reloj-caculadoras, entregadas por ComputerWorld/España a:

—Santiago Casas Duarte. Tarrasa (Barcelona). Aparece revista nº 6.
—Domingo Mánguez Gallego. Madrid. Apareció revista nº 2

4 cintas entregadas por Indescomp a:

—Santiago Martínez Romero. Aparece en revista nº 6
—Domingo Márquez Gallego. Móstoles (Madrid). Apareció revista nº 3
—Luis Casi Gil. Barcelona. Apareció revista nº 4
—José Luis Arostegui Plaza. Granada. Apareció revista nº 5

2 Cintas entregadas por M.I.S.S.A. a:

—David Pérez Corral. Barcelona. Apareció revista nº 3
—Rafael García Segura. Málaga. Aparece revista nº 6

RAMY Y ROMO

1 regla-reloj-calculadora entregada por ComputerWorld/España a:

—María Longa Portabales. Vigo (Pontevedra). Mandó los nombres "EN"-"CHUFE"

1 cinta entregada por M.I.S.S.A. a:

—Belén Longa Portabales. Vigo (Pontevedra). Mandó los nombres "COMMO" - "DORE".

Convocatoria del Primer Concurso Nacional Escolar de Programación patrocinado por Commodore World y Microelectrónica y Control

BASES

1º Podrán participar en el mismo todos aquellos niños y jóvenes que estén en edad escolar pre-universitaria o equivalente.

2º Habrá tres grupos definidos:
a) E.G.B.
b) B.U.P.
c) F.P.

3º Los trabajos pueden ser presentados por equipos o por individuales.

4º Los trabajos presentados por equipos deben especificar:
a) Nombre del equipo.
b) Centro al que representan.
c) Nivel de enseñanza que cursan (E.G.B.-B.U.P-F.P.).
d) Nombre del profesor, monitor o coordinador.

5º Los trabajos presentados por individuales deben ir acompañados de:
a) Nombre del autor.
b) Edad y curso que estudia.
c) (¡) Si concursa representando a un centro: nombre del mismo y nombre del profesor responsable.
(¡¡) Si concursa por sí mismo: nombre del colegio donde estudia y firma del director o profesor responsable avalando que en tanto cuanto conoce al concursante el trabajo que presenta es suyo propio.

6º El tema del concurso es la realización de un programa educativo sobre materia libre a elegir por los concursantes. La extensión del programa es también de libre elección.

7º Cada centro puede presentar el número de equipos o concursantes individuales que desee, siempre que cada trabajo sea enviado por separado.

8º Los trabajos deben incluir:
a) Nombre el programa.
b) Nombre del computador para el que está confeccionado y periféricos adicionales que se necesitan (si fuese el caso).
c) Objetivo del programa.
d) Explicación detallada del programa (a máquina o impresora).
e) Listado completo a impresora.
f) Grabación en cinta o disco.

9º Todos los trabajos presentados deben ser originales e inéditos.

10º Los centros que pueden presentar equipos o concursantes individuales son:
a) Todo colegio, escuela o instituto, público o privado que imparta las enseñanzas de E.G.B., B.U.P, o F.P.
b) Centros de Informática que tengan grupos de enseñanza específica para niños y jóvenes menores de 18 años.
c) Clubs de informática que mantengan actividades específicas para niños y jóvenes menores de 18 años.

11º Los trabajos deben ser enviados antes del 31 de diciembre de 1984.

12º El fallo del concurso se dará a conocer en la primera semana de marzo de 1985 en el lugar, fecha y hora que se indicarán.

13º Los trabajos premiados quedarán propiedad de los patrocinadores.

En nuestro número de septiembre anunciaremos la fecha y lugar donde se celebrará la presentación oficial a la prensa del concurso así como la relación de premios.

Convocatoria del primer concurso nacional de programación en 80 columnas de Commodore-64 y 40-80 columnas del VIC-20 patrocinado por Ferre-Moret, S.A. Con la colaboración de Commodore-World, más de 500.000 ptas. en premios

BASES

1º Podrán participar todos los usuarios de COMMODORE-64 y VIC-20 que hayan obtenido la tarjeta de 80 columnas de FERRE-MORET, S.A.

2º Cada concursante podrá presentar, un programa realizado por él o por un equipo del que deberán indicar un titular.

3º Se enviará la explicación detallada del programa redactado en español, escrito a máquina. El programa se enviará en cassette o disco, adjuntando su correspondiente listado del programa.

4º Se enviarán por correo certificado a la siguiente dirección, FERRE-MORET, S.A. calle Buenos Aires nº 30, 2º-3.ª de BARCELONA-36. Antes del 31 de enero de 1985.

5º Los programas no premiados podrán ser solicitados por sus propietarios en el plazo de treinta días después del fallo.

6º El resultado se comunicará, directamente a los ganadores y se publicará en la revista COMMODORE WORLD.

7º Los premios se entregarán durante la primera semana de marzo de 1985 coincidiendo con la entrega de premios del PRIMER CONCURSO NACIONAL ESCOLAR DE PROGRAMACION.

8º Los programas premiados quedarán en propiedad de FERRE-MORET, S.A.

9º Los programas que no obtengan premio, recibirán, en caso que así lo considere FERRE-MORET, S.A., ofertas de compra de sus programas.

10º Los premios ascenderán a 500.000 Ptas. repartidos de la siguiente manera:

1º 250.000 Ptas.
2º 150.000 Ptas.
3º 100.000 Ptas.

11º Todos los participantes, que no hayan recibido premio u oferta de compra, podrán participar en el concurso de colaboradores que habitualmente COMMODORE WORLD viene realizando con un premio extra de FERRE-MORET, S.A., de 50.000 Ptas.

12º Los trabajos deberán de ser inéditos y desarrollar temas de utilidad, programas profesionales, científicos, educativos, etcétera.

FERRE-MORET, S.A. se reserva el derecho de modificar las bases de este certamen, dejarlo desierto o suprimirlo, por causa de fuerza mayor. En este caso la empresa se compromete a comunicarlo a través de COMMODORE WORLD.

INDICE DE ANUNCIANTES

BOLETIN DE SUSCRIPCION — Commodore World

NOMBRE .... EDAD ....
DIRECCION ....
POBLACION .... (..........) PROVINCIA ....
TELEF. .... MARCA Y MODELO DEL ORDENADOR ....
....
CIUDAD DONDE LO COMPRO .... DISTRIBUIDOR ....
APLICACIONES A LAS QUE PIENSA DESTINAR EL EQUIPO ....
....

Deseo iniciar la suscripción con el nº 7
Adjunto cheque de 2.530 pesetas ☐
Reembolso más gastos del mismo
al recibir el primer nº de la suscripción ☐
(Enviar a la dirección del dorso)
Extranjero $40. Solamente por correo aéreo

Firma,

Desde 1º de julio

DESEO SUSCRIBIRME A **COMMODORE WORLD** POR UN AÑO AL PRECIO DE 2.530 PTS. DICHA SUSCRIPCION ME DA DERECHO, NO SOLO A RECIBIR LA REVISTA (ONCE NUMEROS ANUALES) SINO A PARTICIPAR EN LAS ACTIVIDADES QUE SE ORGANICEN EN TORNO A ELLA Y QUE PUEDEN SER COORDINACION DE CURSOS DE BASIC, INTERCAMBIOS DE PROGRAMAS, CONCURSOS, ETC.

EJEMPLARES ATRASADOS DE COMMODORE WORLD

Commodore WORLD

1	2	3	4	5

Precio por ejemplar 275 ptas.
Forma de pago: sólo por cheque

Peticionario ....
Calle .... **Nº** .... **TELEF.** ....
Población .... **D. P.** .... **Provincia** ....
Incluyo cheque por valor de .... **pesetas + 75 de gastos de envío.**

SERVICIO DE CINTAS

De programas aparecidos en Commodore World.

Título del programa .... publicado en nº ☐
Título del programa .... publicado en nº ☐
Título del programa .... publicado en nº ☐

Precio por cinta 850 pesetas.
* **Precio por disco 1.500 pesetas.**
Gastos de envío 75 pesetas. Forma de pago: sólo por cheque.

Peticionario ....
Calle .... Nº .... TELEF. ....
Población .... D.P. .... Provincia ....

* *Programa evaluación de colegios sólo en discos.*
Incluyo cheque por .... pesetas.
Programa para VIC-20 ☐
Programa para C-64 ☐

EJEMPLARES ATRASADOS DE «CLUB COMMODORE»

Primera época (septiembre-1982 - enero-1984)

Para poder satisfacer la creciente demanda de Club Commodore, agotada en todos sus números, hemos puesto en marcha un Servicio para suministrar fotocopias de los ejemplares que nos sean solicitados.

SERVICIO DE FOTOCOPIAS.— NUMERO DE LA EDICION SOLICITADA.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

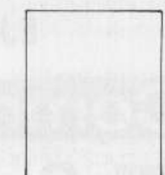
Peticionario: D. ....
Calle .... Nº .... TELEF. ....
Población .... D.P. .... Provincia ....

Forma de pago sólo por cheque
Precio de la edición fotocopiada: 250 ptas.
La colección completa del 0 al 15: 2.500 ptas. + 150 ptas. por gastos de envío.
Incluyo cheque por .... ptas.

Commodore
WORLD

Pedro Muguruza, 4-8º B
Teléf.: 259 54 78
MADRID-16

Sant Gervasi de Cassoles, 39-despacho 4
Teléf.: 212 73 45
BARCELONA-22

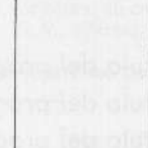


Commodore
WORLD



Pedro Muguruza, 4-8º B
Teléf.: 259 54 78
MADRID-16

Sant Gervasi de Cassoles, 39-despacho 4
Teléf.: 212 73 45
BARCELONA-22

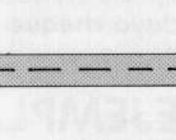


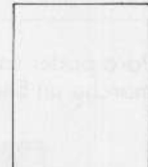
Commodore
WORLD



Pedro Muguruza, 4-8º B
Teléf.: 259 54 78
MADRID-16

Sant Gervasi de Cassoles, 39-despacho 4
Teléf.: 212 73 45
BARCELONA-22





Commodore
WORLD

Pedro Muguruza, 4-8º B
Teléf.: 259 54 78
MADRID-16

Sant Gervasi de Cassoles, 39-despacho 4
Teléf.: 212 73 45
BARCELONA-22

Clave para interpretar los listados

Todos los listados que se publican en esta Revista han sido ejecutados en el modelo correspondiente de la gama de ordenadores COMMODORE. Para facilitar la edición de los mismos en la Revista y para mejorar su legibilidad por parte del usuario, se les ha sometido a ciertas modificaciones mediante un programa escrito especialmente para ello. Para los programas destinados a los ordenadores VIC-20 y COMMODORE 64, en los que se usan frecuentemente las posibilidades gráficas del teclado, se han sustituido los símbolos gráficos que aparecen normalmente en los listados por una serie de letras entre corchetes [] que indican la secuencia de teclas que se deben pulsar para obtener el carácter deseado. A continuación se da una tabla para aclarar la interpretación de las indicaciones entre corchetes:

[CRSRD] = Tecla cursor hacia abajo (sin SHIFT)
[CRSRU] = Tecla cursor hacia arriba con SHIFT)
[CRSRR] = Tecla cursor a la derecha (sin SHIFT)
[CRSRL] = Tecla cursor a la izquierda (con SHIFT)
[HOME] = Tecla CLR/HOME (sin SHIFT)
[CLR] = Tecla CLR/HOME (con SHIFT)

Las indicaciones [BLK] a [YEL] corresponden a la pulsación de las teclas de 1 a 8 junto a la tecla CTRL. Lo mismo sucede con [RVSON] y [RVSOF] respecto a la tecla CTRL y las teclas 9 y 10.

El resto de las indicaciones constan de la parte COMM o SHIF seguidas de una letra, número o símbolo —por ejemplo [COMM+] o [SHIFA]—. Esto indica que para obtener el gráfico necesario en el programa deben pulsarse simultáneamente las teclas COMMODORE (la que lleva el logotipo) o una de SHIFT y la tecla indicada por la letra, el número o el símbolo, en el ejemplo anterior: COMMODORE y + o SHIFT y A, respectivamente.

En los signos gráficos además se cuenta el número de veces que aparece. Por ejemplo, [7 CRSRR] equivale a 7 pulsaciones de la tecla cursor a la derecha y [3 SPC] tres pulsaciones de la barra espaciadora. ■



METEDURAS DE PATA



En el número 5 metimos tres veces la pata:

- **En el programa "ARACNIDA" (pág. 34), aparece "[ERROR]" en las líneas 60 y 20296, en las dos ocasiones debería haber aparecido "[CTRL C]", es decir la tecla de control y la letra C.**
- **En la excursión por la microprogramación (pág. 74), se intercambiaron los términos en uno de los equivalentes BASIC: decía POKE A, 4097 y debía decir POKE 4097, A.**
- **En el anuncio de DELTABIT (Pág. 50) se nos coló el "duendecillo de las erratas" y nos cometió dos. EN la línea donde dice Primer Buffer..., debería de decir Printer Buffer... y en la línea donde dice Wordprocessor+Manuel (casete), debería decir: Wordprocessor+Manual (cassette). Esperamos que en este número el "duendecillo" esté ya con las vacaciones.**



MEA CULPA

Error en el listado del programa de conversión del Kernal 02 a 03

Por aquellas cosas que siempre pasan a pesar de todas las precauciones para que no pasen, el mes pasado se colaron unos errores en la transcripción del programa en BASIC que actualiza la revisión de ROM del COMMODORE 64 (de -02 a-03). Los errores consisten en la desaparición del listado de los signos " " y " ", con lo que ciertas sentencias IF... THEN quedaron totalmente inoperativas, y que las líneas 210 y 220 se mezclaron.

Para redimir este bug, reproducimos a continuación todo el programa de nuevo (esperamos que esta vez salga bien) y pedimos mil perdones por el tiempo que os hayamos hecho perder.

```
100 REM *** COPIA DE ROM A RAM ***
110 READ D:D1=D:CH=0
120 READ X:IF X<256 THEN POKE D,X:CH
=CH+X:D=D+1:GOTO120
130 IF X<>CH THEN PRINT"ERROR[SPC]DE
[SPC]CHEKSUM":STOP
140 SYSD1
150 :
200 REM*** MODIFICAR BYTES ***
210 READ X:IF X<256 THEN POKE D,X:D=
D+1:GOTO 210
220 IFX<>999 THEN D=X:D=D+1:GOTO 210

230 :
300 REM*** ACTIVA RAM ***
310 POKE1,53
320 END
330 :
400 REM*** RUTINA DE COPIA ***
410 DATA 828
420 DATA 160,0,132,2,169,160,133,3,1
77,2,145,2,200,208,249
430 DATA 230,3,165,3,208,1,96,201,19
2,208,238,169,224,208,232
440 DATA 4120
450 :
500 REM*** BYTES A CAMBIAR ***
510 DATA 58540,129
520 DATA 58579,133,169,169,1,133,171
,96,173,134,2
530 DATA 58748,32,240,233,169,39,232
,180,217,48,6,24,105,40,232,16
540 DATA 58763,246,133,213,76,36,234
,228,201,240,3,76,237,230,96,234
550 DATA 58914,145,229
560 DATA 59911,32,218,228,169,32,145
,209,136,16,246,96,234
570 DATA 61332,76,211,228
580 DATA 65408,3
590 DATA 999
```

LOS "STRINGS": CADENA PERPETUA... ¿O NO?

Este artículo te ayudará a desenredar el misterio de los "strings".

Para muchos principiantes, un "string" es simplemente una palabra o frase que se escribe en la pantalla y que luego puede ser olvidada. Sin embargo, el programador con más experiencia se da cuenta de la flexibilidad de los "strings": pueden ser comparados, agregados, clasificados, divididos, medidos y convertidos en número, y gereralmente ayudan a confeccionar un programa más sofisticado.

Comparación de los "Strings"

La comparación de los "strings" se realiza con tanta frecuencia (IF A$=B$ THEN PRINT "CORRECTO!") que probablemente no te has dado cuenta que estás utilizando un símbolo normalmente usado en las matemáticas. Si el signo "igual a" (=) puede ser utilizado con los "strings", también se pueden usar los otros operandos relacionados >, <, <> (mayor de, menor de, no igual a).

El uso de <> es bastante obvio: IF A$<>B$ THEN PRINT "INCORRECTO". Aunque la comprobación de los "strings" para ver si son iguales o no tiene una aplicación evidente para los juegos de pregunta-respuesta, las comparaciones igual/no igual también forman la base de muchas rutinas de búsqueda.

Por ejemplo, un pequeño negocio que tiene una lista de correo de sus clientes. Antes de añadir el nombre de un nuevo pedido a dicha lista, tiene que ser comprobado con los nombres almacenados para evitar una duplicación. Con 100 clientes en un "array", C$, y el nuevo comprador, N$,

```
FOR X=1 TO 100
IF N$    C$(X) THEN NEXT X
```

hará que el ordenador compruebe el nuevo nombre con los de la lista. Si no se encuentra en la lista, se terminará el bucle For-Next y llevará a cabo el siguiente comando —agregar el nuevo nombre a la lista.

Mientras que los signos "igual a" y "no igual a" son bastante sencillos, el uso de "menor de" y "mayor de" con los "strings" resulta técnicamente más complicado. Cuando indiques al ordenador que compare letras, lo que hace realmente es comparar números, utilizando códigos CHR$, así que A<B, K<M, Z>V Y V>R.

El ordenador compara los "strings" mirando las letras una por una, comparando la primera letra de las palabras, a continuación la segunda letra, etc.: ALPHA<BETA; GRAMMA>DELTA. Cuando un "string" se queda sin caracteres, es "menor que" un "string" más largo que empiece por las mismas letras: ASP<ASPEN.

Como te habrás dado cuenta, los operadores de relación son el núcleo de cualquier rutina de clasificación en orden alfabético. La siguiente rutina corta escribirá dos palabras en orden alfabético:

```
10 INPUT A$,B$
20 IF A$<C$ THEN PRINT A$,B$
30 IF B$<A$ THEN PRINT B$,A$
```

Concatenación

Los operadores de relación no se consideran funciones de los "strings", aunque pueden ser utilizados para el manejo de ellos. Otro símbolo matemátco que no constituye una función de los "strings" pero que puede ser utilizado para su manipulación es el signo +.

Los "strings" normalmente no se añaden unos a otros, por lo que esta operación no se llama adición, sino concatenación. La concatenación es simplemente juntar dos "strings": BALON + CESTO resulta en BALONCESTO.

Para demostrar la diferencia entre añadir y concatenar, 123 + 456 = 579, mientras "123" + "456" = "123456". Algunas de las aplicaciones de la concatenación se presentan a continuación en las secciones dedicadas a las funciones de los "strings".

LEN

Una de las funciones de los "strings" más sencillas en los Commodore es LEN, que mide la longitud de un "string". LEN(A$) te proporciona el número de caracteres en A$.

LEN puede ser utilizado para comprobar que la persona usando el teclado esté siguiendo las instrucciones:

```
10 PRINT"ENTRA UNA PALABRA
   DE CINCO LETRAS"
15 INPUT A$
20 IF LEN(A$)<>5 THEN GOTO 15
```



El programa no continuará hasta que se introduzca una palabra de cinco letras.

LEN también puede ser utilizado para centrar una palabra o una frase en la pantalla. La mitad del "string" debe escribirse antes de la columna central. Teniendo esto en cuenta, puedes fijar la TAB según la longitud del "string".

```
10 FOR N=1 TO 6
15 READ A$:PRINT
   TAB(20-LEN(A$)/2)A$
20 NEXT N
25 DATA ESTAS, PALABRAS, ESTAN,
   CENTRADAS, VERDAD?
```

Partición de cadenas ("Slicing")

La partición es la técnica más flexible de todas las de manejo de los "strings". La partición de un "string" significa tomar una parte.

LEFT$ toma la primera parte de un "string", RIGHT$ empieza desde el final, y MID$ toma una parte del centro. Con LEFT$ y RIGT$, tienes que especificar el número de letras que quieres; con MID$, tienes que indicar la posición de la letra inicial, más el número de caracteres que quieres. Si no utilizas un segundo número con MID$, el ordenador supone que quieres el resto del "string". Por ejemplo:

```
A$=REACCION
LEFT$(A$,5)=REACC
RIGHT$(A$,2)=ON
MID$(A$,3,3)=ACC
MID$(A$,3)=ACCION
```

Prueba la combinación de partición y concatenación:

```
10 A$="GODSON":  B$="GRAND-
   FATHER"
15 C$=LEFT$(A$,3)+RIGHT$(B$,6)
20 D$=LEFT$(B$,5)+RIGHT$(A$,3)
25 PRINT A$,B$,C$,D$
```

La participación es una herramienta enormemente útil. Para catalogar los libros de tu biblioteca personal, podrías utilizar un "string" que incluye el autor, un título abreviado y un código de dos letras para el tipo de libro: ciencia ficción, clásico, ciencia informática, novelas, etc.

Asimov, Foundation-SF
Twain, Conn, Yakee-CL
Harrison, Byteing-CS
Ludlum, Matarese-NV

Mediante el uso de RIGHT$(B$,2), le indicas al ordenador que mire solamente el tipo de letra; la incorporación de esto en una rutina de clasificación te proporciona un inventario por categorías.

La partición de los "strings" proporciona unos efectos interesantes en pantalla, como la representación de una palabra letra por letra. (Obviamente, siempre funciona así, pero demasiado de prisa para que tú lo veas.) Prueba lo siguiente:

```
10 A$="DESPACIO"
20 PRINT LEFT$(A$,1); GOSUB 80
30 FOR L=2 TO 7
40 PRINT MID$(A$, L,1);:GOSUB 80
50 NEXT L
60 PRINT RIGHT$(A$,1):END
80 FOR T=1 TO 750: NEXT:RETURN
```

La línea 20 escribe la primera letra en la pantalla; la línea 40 escribe cada una de las letras siguientes; y la línea 60 escribe la última letra. Los puntos y comas detrás de los paréntesis en las líneas 20 y 40 mantienen la posición de las palabras escritas para que las letras se escriban en la misma línea. La subrutina es simplemente un bucle de retraso.

VAL

Los números dentro de los "strings" tienen que ser convertidos en "valores" antes de poder ser usados en cualquier operación matemática. La función VAL proporciona el valor de cualquier número dentro de un "string", siempre que este no vaya precedido por un símbolo no numérico.

Aunque los "strings" puedan ser concatenados, no pueden ser sumados. La función VAL te permite realizar sumas (o cualquier otra operación matemática) con los "strings" numéricos. Considera lo siguiente:

```
A$="42":B$="24"
A$+B$="4224"
VAL(A$)+VAL(B$)=66
```

Te podrías estar preguntando por qué un número tiene que estar dentro de un "string". Por ejemplo, la fecha de un cumpleaños que se almacena de forma numérica a veces necesita un cero inicial. Por ejemplo, 12341 significa el 23 de enero o el 3 de diciembre, 1941? El ordenador no almacenará 012341 como número, pero sí almacenará "012341" como un "string".

Tal vez necesites saber la edad que tendrá una persona en el año 1993, cuando el plan de jubilación de la empresa será sometido a una modificación radical. 93-VAL(RIGHT$(A$,2)) te indicará que dicha persona tendrá 52 años.

STR$

La función STR$ complementa la función VAL, dado que convierte un número en un "string".

Se puede usar para consolidar la información en un "string" fácil de almacenar. Si tienes un "array" de "strings" (S$) que contiene los periféricos que te has comprado, y un "array" numérico (N) que contiene su precio, podrías convertir esta información en un sólo "string":A$(1)=S$(1)+STR$(N(1))

El "array" A$ podría parecerse a lo siguiente:

```
A$(1)="ORDENADOR 80.000"
A$(2)="UNIDAD DE DISCO 95.000"
```

etcétera.

Se utiliza STR$ porque a menudo los "strings" son más fácilmente manipulados que los números, en la partición, por ejem- ▶

Listado del juego "Desorden", una muestra de las funciones de los "Strings".

```
10 PRINT"[CLR]"
20 PRINT"ENTRA[SPC]UNA[SPC]PALABRA[SPC]
DE[SPC]5[SPC]A[SPC]10[SPC]LETRAS"
30 INPUTA$
40 IF LEN(A$)<5 OR LEN(A$)>10 THEN P
RINT"HE[SPC]DICHO[SPC]DE[SPC]5[SPC]A
[SPC]10[SPC]LETRAS":GOTO30
50 B$=A$:LA=LEN(A$):LB=LEN(B$):C=0
60 PRINT"[CLR]"
70 FOR N=1 TO LA
80 R=INT(RND(0)*LA)+1
90 S$=S$+MID$(A$,R,1)
100 A$=LEFT$(A$,R-1)+RIGHT$(A$,LA-R)

110 LA=LA-1:NEXT N
120 PRINTS$:PRINT
130 PRINT"PULSA[SPC]'I'[SPC]PARA[SPC]
INTENTAR,[SPC]'A'[SPC]PARA[SPC]AYUDA
":PRINT
140 GETP$:IFP$=""THEN140
150 IFP$="I"THEN180
160 C=C+1:IF C=LB-1THEN200
170 PRINTS$,,LEFT$(B$,C):PRINT:GOTO1
30
180 PRINT"CUAL[SPC]ES[SPC]TU[SPC]INT
ENTO?":INPUTG$:IFG$<>B$ THEN PRINT"N
O":PRINT:GOTO130
190 PRINT"ACERTASTE"
200 PRINT"LA[SPC]PALABRA[SPC]ERA[SPC]
"B$
```

plo, o en el centrado u otras alineaciones. Probablemente te has dado cuenta que el ordenador alínea los números en el margen izquierdo.

```
43271
36
864
762895
```

Si estos números se colocasen en unos "strings", podían ser fácilmente justificados en el margen derecho. El siguiente programa convierte los datos numéricos en "strings" y a continuación los manipula de una forma semejante a la de la rutina de centrado explicada antes.

```
10 FOR N=1 TO 10
20 READ A:A$=STR$(A)
30 PRINT TAB(20-LEN(A$))A:NEXT N
40 NEXT N
50 DATA 34, 5678,39,7,12345,7431,
   8754342,42,9215,6
```

Probablemente te has dado cuenta de que sería más fácil asignar cada dato directamente a A$ e ignorar STR$. La elegancia no era el objetivo principal de este programa ejemplo; simplemente sirve como una demostración de la función STR$. El uso más frecuente de la conversión STR$ sería para alinear unos números que tienen asignados unas variables numéricas, tal vez como parte de un "array", para que puedan ser manipuladas de forma numérica cuando sea necesario.

Juegos

La partición de los "strings" se adapta fácilmente a los juegos de palabras. "Desorden" es un programa que hace uso de todas las funciones de los "strings" excepto STR$, además de la concatenación. Incorpora una rutina de "desordenar" diseñada por Dam Tookmanian.

"Desorden" es para dos jugadores. El primero entra una palabra que el ordenador desordena y saca en pantalla; el segundo jugador tiene que adivinar la palabra. Puede pedir ayuda, y el ordenador le indicará letras sucesivas de la respuesta hasta que haya enseñado todas menos las dos últimas.

Dado que este juego sirve solamente para demostrar las funciones de los "strings", constituye una versión bastante simplificada de un juego de ordenador —sin gráficos, ni sonido, ni siquiera la puntuación! Lee la explicación del programa y añade las líneas más apropiadas para que quede más completo —la puntuación debe coincidir con el valor de C, que cuenta el número de pistas proporcionadas.

La línea 40 comprueba si la palabra introducida es de la longitud correcta. La línea 50 asigna el mismo valor a B$ que a A$, dado que A$ quedará destrozada durante el procedimiento de desordenar la palabra y no podrá dar la respuesta correcta.

Por el mismo motivo, se necesita LB para representar la longitud de la palabra introducida; LA será modificada durante el proceso de desordenar. C servirá como contador para el número de veces que se pide ayuda.

Las líneas 70-110 son las rutinas de desordenar. Primero, se elige un número aleatorio en algún sitio dentro de la longitud de A$. La letra que ocupa la posición A$ se coloca en S$, la palabra desordenada.

En la línea 100 la letra que ha pasado a S$ se quita de A$ al volver a definir A$. Ahora A$ consiste en todo hasta la letra quitada, más todo lo que vaya detrás. LA se decrementa por uno porque A$ se queda más corta por un carácter, y otro número aleatorio tiene que ser elegido dentro de los límites correctos.

Este procedimiento se repite hasta que todas las letras hayan sido transferidas al azar de A$ a S$. (Ten en cuenta que el bucle For-Next está fijado para igualar el número de caracteres en A$.) Ahora A$ es un "string" vacío, y lo que diga el jugador será comparado con B$.

Si el jugador pide ayuda, C se incrementa por uno en la línea 160, y si se da ayuda tantas veces de modo que sólo quedan dos letras sin descubrir, el juego termina. Dado que C se incrementa en esta línea, LB-1 comprueba si faltan dos letras. Fíjate en el uso de LB en vez de LA, que iguala cero después de la rutina de desordenar.

En la línea 170, B$ se divide para que sea igual al número de veces que se ha pedido ayuda hasta el momento. Cada vez que un jugador intenta adivinar un número, este (G$), se compara con B$. Una respuesta incorrecta vuelve a enviar el programa a la línea 130, mientras que una respuesta correcta es reconocida. ■

NOTICIAS COMMODORIANAS

¡Un respeto al público, por favor!

Sinclair tira para casa...

...Y esto nos parece muy bien, muy bien, siempre y cuando no se perjudique a otras marcas informando mal al gran público. Esto es exactamente lo que pasó el domingo, día 17 de junio por la mañana en un programa de radio (Radio Cadena E., 88.200 KHz FM) patrocinado por Investrónica. A continuación presentamos una transcripción de parte del programa seguida de nuestra réplica y que nuestros lectores juzguen por sí mismos.

Programa "Sólo Ordenadores"

"Sobre la consulta de un oyente invidente:

...Quería saber algo que pudiese conectar a su C-64 para que la salida de pantalla se pudiese escuchar simultáneamente por el altavoz del monitor.
RESPUESTA: En este caso tenemos que decirle que tal y como le habíamos comentado a el y cualquier otra persona que se interese sobre el tema, existe un sintetizador de voz comercializado por Commodore en forma de disco, es decir, es un programa que tiene ...eh... bueno... como utiliza simplemente los generadores de ruido y de sonido de que dispone el Commodore 64, lógicamente no se consigue un alto nivel de inteligibilidad, aunque naturalmente ...eh... la calidad ...eh... bueno, pues es suficiente para determinado tipo de aplicaciones. No es este el caso en que el oyente nos comentaba que se le quedaba demasiado corto en cuanto a prestaciones para lo que el quería, y le ofrecimos hacer una gestión en Inglaterra por medio de un compañero que marchaba para Londres y que ha regresado a lo largo de esta semana... eh... tenemos que ...en fin, lamentamos tener que informarle que no hemos conseguido una información... digamos definitiva, parece que todo se queda en rumores. Hay algunas casas que han construido sintetizadores de voz, pero curiosamente están más enfocados hacia el Sinclair Spectrum que hacia el Commodore 64. La única información que podemos darle, como digo, es la de estas casas, y... bueno, como ya no coincide exactamente con lo que nos pedía, si realmente sigue interesado en el tema, pues podemos darle la dirección de la gente en Inglaterra, aunque insistimos, ellos no tenían nada para el Commodore 64, al menos esta es la información que hemos podido recoger a través de este compañero, que como digo marchó para Londres".

(transcripción íntegra de la grabación realizada en magnetofon por Diego de "Commodore World").

Deseamos formular nuestra más enérgica protesta

La primera obligación de todo periodista es proporcionar una información clara, concisa y bien documentada, si no es así, se presta a todo tipo de interpretaciones y malentendidos.

Nosotros no hemos tenido que fiarnos de las pesquisas de ningún compañero en Londres, nos ha sido muy fácil enterarnos de marcas, distribuidores y precios tanto en Inglaterra como en los Estados Unidos de sintetizadores de voz.

Por ejemplo... concretamente en Inglaterra existe un sintetizador con el nombre comercial de "Speakeasy" que sale muy bien de precio (30 libras) e incluye un vocabulario sin límites mediante el uso de alófonos, resulta fácil de usar tanto en programas escritos en Basic como en programas escritos en código máquina, dispone de una fuente de energía directa del mismo ordenador, altavoz incorporado, y se adapta a cualquier marca de ordenador simplemente cambiando el cable.

Y por ejemplo... en los Estados Unidos (fabricado por el Alien Group) tenemos el "Voice Box" para el C-64 y el VIC-20. Aparte de incluir todas las características de un sintetizador de voz, dispone de un programa opcional que convierte el C-64 en un sintetizador de música.

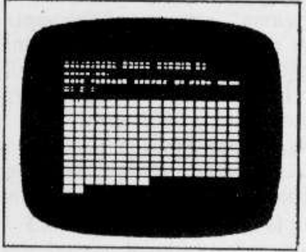
Y por ejemplo... la misma casa Commodore comercializa el "Magic Voice" y el disco "Magic Voice Vocabulary" que contiene un vocabulario de 10.000 palabras y capacidad para almacenar palabras del mismo usuario. Aunque todavía no están a la venta en España, no existe ningún problema en pedirlos.

Y estos son sólo algunos ejemplos sacados a toda velocidad al cierre de la revista (ver fecha del programa más arriba). Lo que queda claro es que "intereses" hay de todos tipos: económicos, creados y adquiridos, pero en cualquier caso exigimos una ética y el derecho de réplica.

Que conste que nuestro interés es solamente el que debe ser obligatorio para cualquier medio de comunicación: el interés de informar.

¡ATENCION!

¡INCREIBLES TARJETAS PARA COMMODORE 64 Y VIC-20 DE 80 COLUMNAS, 40/80!



80 columnas

- Tarjeta VIC-20 40/80 columnas
- Tarjeta COMMODORE 64 80 columnas
- Tarjeta VIC-20 Ampliación de memoria 64 K - RAM
- Slots de expansión para VIC-20 y 64 de 2 y 5 para conexión de tarjetas
- Cuidado con las imitaciones. Sólo son nuestras tarjetas si van debidamente protegidas con caja y garantizadas.

DISTRIBUIDORES EXCLUSIVOS PARA TODA ESPAÑA

PEDIDOS DISTRIBUIDORES COMMODORE

FERRE-MORET, S. A.
Sección Hardware y software
Tel. 93/ 250 84 40/ Contestador Aut.
Escribir At. Director Comercial
C/. Buenos Aires n.º 30 2º 3ª
BARCELONA-36

PEDIDOS USUARIOS

FERRE-MORET, S. A.
Tel. 93/ 250 84 40/ Contestador Aut.
Escribir indicando la sección
C/. Buenos Aires n.º 30 2º 3ª
BARCELONA-36

¿RIESGOS?... NO GRACIAS

No te arriesgues más con programas importantes.
Este programa para copiar programas evita los riesgos y da más seguridad.



Los usuarios de la unidad de discos 1541 saben que algunos programas no pueden ser copiados en otros discos utilizando los métodos normales. El programa utilitario "Copiador" te ayudará a solucionar este problema.

Un programa escrito en Basic puede ser cargado de disco y se puede volver a salvar en un disco diferente insertado en la unidad. Además, esto puede funcionar con algunos programas escritos en lenguaje máquina, pero sólo con algunos —y no con ficheros de datos—.

Muchas veces los usuarios que se acaban de comprar una unidad de discos 1541 intentan copiar el programa DOS WEDGE que viene gratis con la unidad. Este "wedge" viene en dos partes: el programa mismo en lenguaje máquina (nombre de fichero DOS 5.1.) y un corto programa cargador escrito en Basic (nombre de fichero C-64 WEDGE o DOS WEDGE 64).

Es fácil copiar en disco el programa cargador y cualquier programa escrito en Basic pero no se puede utilizar el mismo método para copiar programas escritos en lenguaje máquina, y esto es precisamente lo que hace el "copiador".

Maniobras de Memoria

La idea general de la operación del "copiador" es muy sencillo. Primero el programa se copia desde el disco en un "array" de memoria. Se inserta el nuevo

VIC 20 ó C-64
UNIDAD DE DISCOS

```
100 DIMA$(150)
200 PRINT"[CLR]PROGRAMA[SPC]COPIADOR
"
210 PRINT
300 INPUT"INSERTA[SPC]DISCO[SPC]ORIG
INAL,RETURN";A$
310 PRINT
400 INPUT"NOMBRE[SPC]DE[SPC]FICHERO"
;N$
410 INPUT"TIPO[SPC]DE[SPC]FICHERO[SPC]
(S,P,U)";F$
420 PRINT
1000 OPEN15,8,15
1100 OPEN2,8,2,"0:"+N$+","+F$+",R"
1200 GOSUB6000
2000 I=1:J=0
2100 GET#2,A$
2200 IFA$=""THENA$=CHR$(0)
2300 A$(J)=A$(J)+A$
2400 IFST<>0THENCLOSE2:GOTO3000
2500 I=I+1:IFI>254THENI=1:J=J+1:PRIN
TJ;
2600 GOTO2100
3000 PRINT
3010 PRINT
3100 INPUT"INSERTA[SPC]DISCO[SPC]DES
TINO,RETURN";A$
3200 PRINT#15,"I"
4000 OPEN2,8,2,"@0:"+N$+","+F$+",W"
4100 GOSUB6000
5000 FORX=0TOJ
5100 PRINT#2,A$(X);
5200 NEXTX
5300 CLOSE2
5400 CLOSE15
5500 END
6000 INPUT#15,EN,EM$,ET,ES
6100 IFEN=0THENRETURN
6200 PRINT"ERROR[SPC]EN[SPC]DISCO"
6300 PRINTEN,EM$,ET,ES
6400 CLOSE2
6500 CLOSE15
7000 REM
7001 REM          JOHN OLSEN
7002 REM          P.O. BOX 181
7003 REM          NEWBERG, OR 97132
```

Listado del programa "Copiador"

disco y el contenido del "array" del "copiador" puede contener hasta 150 bloques.

Este programa funciona tanto en el VIC-20, como en el C-64, pero dado que el VIC dispone de menos capacidad de memoria, solamente programas cortos pueden ser copiados. Para poder ejecutarlo en el VIC-20, el tamaño del "array" en la línea 100 tiene que ser modificado para que no resul te en un error "out-of-memory".

Detalles del programa

La línea 100 reserva espacio para 150 bloques en el "array" A$. (Cada bloque contiene 254 bytes de datos del programa). El programa que se está copiando quedará almacenado en este "array" mientras se intercambian los discos en la unidad. Las líneas anteriores a 1000 no necesitan ninguna explicación.

Las líneas 1000 y 1100 abren el canal de comandos a la unidad y abren el fichero que será copiado. La línea 1200 comprueba si existe algún error en disco.

Las líneas dos mil leen el programa byte por byte y lo colocan en bloques de 254 bytes, almacenados en el "array" A$. La línea 2100 lee el byte. La línea 2300 lo añade al bloque en el "array".

Tal vez este programa utilitario resulte un poco lento, pero evita problemas al copiar programas.

La línea 2400 comprueba el estado para ver si se ha llegado al final del programa. Si se ha llegado al final del programa, el fichero se cierra. Si no, se incrementa el contador I. Si el bloque está lleno se inicia un bloque nuevo.

La línea 2200 se encarga de un caso especial leyendo un byte del disco. Si el byte es un cero, la sentencia Get 2 lo lee como un "string" nulo (vacío). Cuando este se vuelve a escribir en disco, la sentencia Print#2 no escribe nada, y tu problema pierde de repente un byte importante.

Para corregir este problema, cualquier cero se convierte en un "string" de caracteres de cero cuando se escribe en el bloque. Por lo tanto, cuando se vuelve a escribir en disco, se escribe el cero.

Las líneas de tres mil le indican al usuario que intercambie los discos y que inicialice el disco nuevo. Si quieres copiar el programa en el disco nuevo bajo un nuevo nombre de fichero, se inserta la línea:

```
3300 INPUT"NUEVO NOMBRE DE FICHERO"; N$
```

De esta forma se puede utilizar un nuevo nombre de fichero. Si no hace falta cambiarlo, se pulsa la tecla "Return".

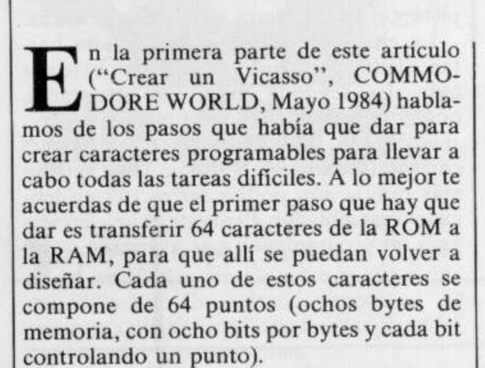
Las líneas de cuatro mil abren el fichero que será escrito y comprueban cualquier error de disco.

Las líneas de cinco mil son las que copian el programa en disco. Cada bloque se lee del "array" y se escribe en disco en el mismo orden en que se ha leído esto se realiza con un bucle For/Next. A continuación se cierra el fichero —junto con el canal de comandos— y el programa termina.

Las líneas de seis mil son las subrutinas que comprueban los errores de disco. El programa comprueba el canal de comandos, y si no encuentra ningún error (EN=0), vuelve al programa principal. De no ser así, imprime el número del error, el mensaje, la pista y el sector, cierra el fichero y se acaba.

La ejecución del programa resulta bastante lenta, dado que tiene que leer byte por byte. La segunda mitad del programa, donde se realiza la copia, funciona mucho más deprisa, dado que se escriben bloques enteros (sectores del disco). Una versión más rápida aún podría ser escrita en lenguaje ensamblador. Pero este es un ejercicio para un programador más avanzado. ■

SEGUIMOS CON LOS VICASSOS



Entra en el mundo multicolor de los caracteres gráficos con las técnicas detalladas en este artículo

**VIC-20
NO AMPLIADO**

En la primera parte de este artículo ("Crear un Vicasso", COMMODORE WORLD, Mayo 1984) hablamos de los pasos que había que dar para crear caracteres programables para llevar a cabo todas las tareas difíciles. A lo mejor te acuerdas de que el primer paso que hay que dar es transferir 64 caracteres de la ROM a la RAM, para que allí se puedan volver a diseñar. Cada uno de estos caracteres se compone de 64 puntos (ochos bytes de memoria, con ocho bits por bytes y cada bit controlando un punto).

Los caracteres multicolores también se componen de 64 puntos, pero cuando se escriben, sólo son visibles 32 puntos más grandes. Parece que el ordenador dispone de un método para saber los colores que tiene que utilizar, y esto lo hace agrupando los puntos en parejas. Se necesitan dos puntos de alta resolución para componer un punto multicolor;

Haz la prueba introduciendo POKE 646,10 y a continuación teclea unas cuantas letras. Las letras parecen extrañas y son difíciles de leer dado que cada punto es el doble de ancho y son de diferentes colores. Ahora pulsa las teclas CTRL y 7 para que la pantalla se vuelva a su estado normal.

Una forma de entrar en el mundo de los caracteres multicolores es de introducir mediante un Poke en la posición 646 un número igual al Poke normal del color de caracter más 8. De esta forma se puede imprimir en la modalidad multicolor. Los colores normales pueden ser restaurados mediante el uso de la tecla CTRL y las teclas de color de forma normal.

Los caracteres multicolores pueden utilizar hasta cuatro colores: la pantalla, el borde, los colores auxiliares y de los caracteres. Se fijan los colores para la pantalla y el borde introduciendo los valores encontrados en la página 134 del "Manual del Usuario" en la posición 36879 mediante un Poke. El color auxiliar, utilizado solamente para los caracteres multicolores, también se fija para la pantalla entera mediante el uso de la siguiente fórmula, donde X es igual al Poke de color de la figura 1.

0 negro	**8 naranja**
1 blanco	**9 naranja claro**
2 rojo	**10 rosa**
3 cian	**11 cian claro**
4 púrpura	**12 púrpura claro**
5 verde	**13 verde claro**
6 azul	**14 azul claro**
7 amarillo	**15 amarillo claro**

Fig. 1. Pokes de color para el VIC-20.

00=color de pantalla.
01=color del borde
10=color del carácter
11=color auxiliar.

Fig. 2. La combinación de los bits para representar los colores.

Poke 36878, X*16

El color de carácter es el único que puede ser diferente para cada carácter. Se fija de la misma manera descrita antes, o el color o código de carácter normal más 8 puede ser introducido en la posición de color de pantalla mediante un Poke.

La figura 2 muestra la combinación de bits necesaria para representar cada color. Para demostrar cómo funciona esto, vamos a formar tres caracteres con los ejemplos de la Figura 3.

Si dispones de un Generador de Caracteres Programables, lo puedes emplear para dibujar estos caracteres. Si no, puedes teclear el programa corto del listado 1. Al ejecutar cualquiera de los dos programas, pulsa la tecla @ y se producirá el primer carácter de la Figura 3. El carácter ha sido dividido en cuatro cajas pequeñas, cada una conteniendo el modelo de bits para un color diferente.

Ahora se introduce POKE 646,13 y se teclea @. De repente los huecos del carácter han sido rellenados, y ahora se compone de cuatro bloques sólidos de color. En este caso, el color auxiliar es negro, el color del carácter es verde, el borde es cian y la pantalla es blanca. Ahora teclea:

```
POKE 36878, 10*16: POKE 646, 13;
POKE 36879, 24
PRINT CHR$(65): PRINT CHR$(66).
```

Los caracteres A y B han sido transformados en el perfil de una niña.

El detalle no es muy bueno dada la baja resolución. Sin embargo, haciendo una

@	A	B
11111010	00000000	00100000
11111010	00000000	00100000
11111010	00010100	00100000
11111010	00010100	00100000
01010000	00110100	00100000
01010000	00110100	00111100
01010000	00110101	00111100
01010000	00110101	00110000

Fig. 3. Tres caracteres de color.

combinación de colores y utilizando el color auxiliar rosa para la piel (un color que antes no podía haberse utilizado para los caracteres), se crea una figura atractiva que quedaría bien en un programa de juegos.

Una ventaja del color auxiliar es que permite la introducción de colores que se encuentran fuera de la gama de colores de los caracteres estandard. Desafortunadamente, sólo se puede utilizar un color auxiliar para la pantalla entera a la vez.

Como podrás ver, el truco de dibujar los caracteres multicolores utilizando el Generador de Caracteres Programables es aprender a visualizar cómo aparecerán los caracteres después de haberles convertido en caracteres multicolores —con los puntos dobles convertidos en puntos multicolores sólidos.

Si esto resulta ser demasiado difícil, otro método es dibujar el carácter como un carácter sólido, siempre manteniendo los puntos en parejas. Al completarse el carácter, se utiliza la función de borrar de la tecla F3 para crear el correcto modelo de puntos para los colores. Se realiza la prueba introduciendo el Poke de color para el carácter multicolor deseado en la posición 646 y pulsando a continuación la tecla CTRL y las teclas azules para que la pantalla se vuelva a su estado normal antes de continuar.

Prueba una cosa más, tecleando lo siguiente:

```
PRINT CHR$(147): POKE 646,10:
FOR T=0 TO 220: PRINT CHR$(64);;
NEXT.
```

Ahora introduce los siguientes Pokes, uno por uno.

```
POKE 36878, 2*16
POKE 36879, 25
POKE 36878, 9*16
```

En cada caso la pantalla se convierte en seguida en lo que parece ser un carácter totalmente distinto. Esta acción instantánea por toda la pantalla constituye otra ventaja de los caracteres multicolores.

Los efectos especiales que pueden ser utilizados para juegos sólo son limitados por tu misma imaginación. ■

ADVERTENCIA: El Programa que aparece a continuación debe añadirse al programa "Crear un Vicasso" publicado en el número 4 de "Commodore World".

```
10 POKE 52,28:POKE56,28:CLR
20 FOR T=7168 TO 7679:POKE T,PEEK(T+
25600):NEXT
30 POKE36869,255
40 READ A: IF A=-1 THEN200
50 FOR T=0T07:READ B:POKE7168+A*8+T,
B:NEXT
60 GOTO 40
61 DATA 0 ,250,250,250,250,80,80,80,
80
62 DATA 1 ,0,0,20,20,52,52,53,53
63 DATA 2 ,32,32,32,32,32,60,60,48
200 DATA-1
```

MUSICA MAESTRO

Aunque tú no seas músico, puedes convertir tu C-64 en maestro con este programa musical



El programa "Música Maestro" hace que el Commodore 64 produzca música. Este programa facilita la tarea de añadir melodías y canciones a tus propios programas.

Si no sabes nada sobre el interface de sonido (SID), Voz 1, 2 ó 3 la diferencia entre C# y G, aquí tienes la solución.

Si estas cosas tampoco te importan mucho y sólo te interesa que producir el sonido, este programa estaba escrito para tí.

Pero no te sorprendas si acabas aprendiendo bastante sobre el SID.

Teclea el programa tal y como viene en el listado (Ten cuidado de no omitir los puntos y comas). Salva el programa antes de ejecutarlo.

Ahora ejecuta el programa. Primero aparece la página del título y alguna información sobre el programa. Pulsa la barra del espacio para empezar.

La siguiente pantalla te da la oportunidad de elegir y definir tu propia voz con: volumen, forma de ondas, ataque/decaimiento, sustento/caída y la duración de la nota. Pulsa la tecla "Return" y el programa la asignara los valores preseleccionados que luego puedes modificar para experimentar, si quieres.

A continuación se presenta la pantalla de prácticas. Cuando pulsas la tecla A, la letra aparecerá en pantalla, y escucharás la nota musical A (octava 4). Cuando pulsas shift/A, escucharás A# y ambos caracteres aparecerán en pantalla. Lo mismo ocurre con A-G. Todas las otras teclas son ignoradas excepto la P, las teclas de funciones (F1-F8), la tecla INST/DEL, la tecla "Return" y la tecla de flecha arriba.

La P es para hacer una pausa; cuando aparece no se oye ningún sonido.

La tecla flecha arriba borra la pantalla. (Puedes jugar con la pantalla entera, pero la zona encima de la línea central se reserva para prácticas).

La tecla "Return" inicia otra línea de notas.

Pulsando la tecla INST/DEL dos veces se borra cada nota, una por una.

F3 borra la pantalla de prácticas e introduce el programa Música Maestro. Se borra toda la pantalla excepto las notas en la zona de Prácticas. Se presente un nuevo menú, ofreciendo unas opciones adicionales.

F7 vuelve al principio donde tu seleccionas y defines la forma de las notas.

Juega con la pantalla de prácticas todo el tiempo que quieras. Cuando consigues una canción o una serie de notas que te gustan, pulsa F3 y pasarás a la pantalla de opciones de imprimir y tocar.

Imprimir y tocar

Aquí es donde se realiza todo el juego y el trabajo. Ahora la pantalla debe presentar:

F1 SOLO DATOS
F2 IMPRIMIR TODO
F3 PRACTICAR
F4 IMPRIMIR NOTAS
F5 TOCAR MELODIA
F6 DURACION
F7 VOLVER A EMPEZAR
P PAUSA ↑ BORRAR TODO

Las notas que queden por encima de la línea media (de la pantalla de prácticas) deben estar todavía en pantalla, y un pequeño símbolo de "mayor que" debe salir disparado por la pantalla, borrando las dos líneas por debajo de la línea media.

Tus notas estarán colocadas en un "array" que fue dimensionado en la línea 15 para contener un máximo de 200 notas. La mitad inferior de la pantalla puede representar unas 200 notas. Durante este tiempo, copia tus notas de práctica o toca otra cosa.

Si quieres volver a empezar, pulsa la tecla flecha arriba, y todas las notas que acabas de tocar (excepto las notas en la zona de prácticas) serán borradas. Si pulsas F3, borrarás todo y volverá la pantalla de prácticas.

Después de componer una melodía en pantalla, pulsa F5, y tu melodía sonará como si fuese parte de un programa. (Nota: C, C, C, hará sonar la C para una duración de tres notas. Si quieres escuchar la nota C tres veces distintas, toca C P C P C).

Cuando funciona, suena

Cuando tu melodía suena bien, pulsa F2 (asegúrate de que la impresora esté preparada) y el programa imprimirá el listado total del programa. Para utilizarlo lo único que tienes que hacer es copiarlo en un programa. Todo se encuentra allí —cada Poke, cada Read, cada bucle For...Next y cada dato.

Sin embargo, a lo mejor tendrás que hacer alguna modificación al copiar los datos. El programa lista todos los datos en una sola línea. Si tu melodía requiere más de una línea de datos, simplemente añade otro número de línea de datos y continúa.

F1 imprime solamente los datos necesarios para la melodía. Más tarde, tú probablemente elegirás esto con más frecuencia (para ahorrar papel).

F4 imprime las notas/letras. (Música Maestro) utiliza la octava 4). Al utilizar estas notas, podrías consultar la página 152 del "Manual del Usuario del Commodore 64" para traducir cada nota en la frecuencia alta y baja para otra octava distinta.

El ordenador te pide el nombre de cada melodía que se imprime. Ahora la impresora lleva el control. Cuando termina la impresora, también imprimirá la cuenta total de los datos. A lo mejor esto te resulta necesario si utilizas otras sentencias Read en un programa.

F6 te permite elegir otra duración para la melodía sin tener que salir de esta modalidad. Nota: INST/DEL no fue incluido en la modalidad Imprimir y Tocar dado que las notas están colocadas en un "array". Si cada nota utilizara solamente un caracter, INST/DEL podía haberse incluido. Pero, por ejemplo C# ocupa dos caracteres. Borrarla hubiera estropeado la cuenta de los "arrays".

Espero que disfrutes con Música Maestro y que proporciona mucho sonido a tus programas.

Aquí presento unas melodías para que experimentes:

—FPFPFPCCPDPDPCCPAPAGPGPFF
—GEGPGEGPPAPGPFPEPDPEPFPEF
GPCPCPCPCPDEFGPG

Después de familiarizarte con el programa, prueba las mismas melodías con distintas formas de ondas, ataque/decaimiento, sustento/caída y duración. ■

C.-64



LA CONFIANZA IMPULSA NUESTRO CRECIMIENTO

Excelentes resultados económicos, colaboración en diversos frentes de interés comunitario y progresiva actividad de la Obra Social.

(Información sobre el ejercicio de 1983, aprobado por la Asamblea General celebrada el día 15 de junio de 1984.)

GESTION ECONOMICA

La confianza aumenta los depósitos

La confianza de los clientes de "la Caixa" tiene una repercusión directa en el crecimiento de los recursos captados, que durante 1983 ha sido de 151.774 millones de pesetas.

Esta cifra equivale a una tasa de crecimiento del 20,5 %, cifra superior a la media de todo el conjunto de Cajas y Bancos. Contando este incremento, a finales de 1983 el total de recursos ajenos ascendía a 890.908 millones de pesetas, a los que hay que añadir el importe de los activos intermediados, que totalizaban la cifra de 46.912 millones de pesetas.

Ampliación de las modalidades de ahorro

En 1983 la ampliación de las modalidades de ahorro de "la Caixa" ha permitido ofrecer a sus clientes muchas posibilidades de inversión: desde la popular libreta a la vista hasta Certificados de Depósito, Cédulas Hipotecarias, Bonos de Tesorería, Pagarés del Tesoro, Pensiones Temporales y Vitalicias para la Tercera Edad, Libreta Agrícola y Ahorro Vivienda, entre otros.

450 millones diarios en nuevos créditos

La actividad crediticia de "la Caixa" durante 1983 ha sido muy intensa. El total de créditos concedidos en este período ha sido de 136.000 millones de pesetas, lo que representa un ritmo de 450 millones por día hábil y una tasa de crecimiento de la cartera del 22,3 % respecto al año anterior.

La colaboración con las diferentes Administraciones ha representado la concesión de créditos por un valor superior a los 17.000 millones de pesetas.

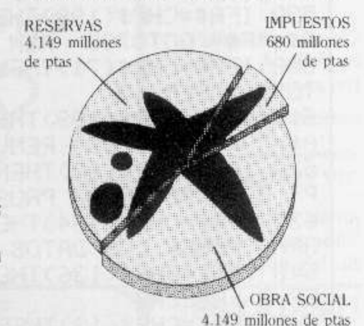
Expansión de la red de oficinas

En 1983 se han inaugurado 39 nuevas oficinas, que sumadas a las ya existentes, hacen un total de 836. En conjunto, y considerando también los cajeros automáticos, "la Caixa" dispone de la red de oficinas más extensa de Cataluña y Baleares.

En este apartado destacan las 10 oficinas Singulares y la presencia en Madrid, Bilbao, Zaragoza y Valencia.

Los beneficios y su distribución

El ejercicio de 1983 ha dado un beneficio neto de 8.978 millones de pesetas, exactamente 2.548 millones más que el año pasado. Estos beneficios se han distribuido de la siguiente manera:



Con el incremento de las reservas que resulta de esta distribución de los beneficios y considerando el efecto de actualización del patrimonio inmobiliario, los Recursos Propios de "la Caixa" han alcanzado la cifra de 106.331 millones de pesetas, lo cual permite que "la Caixa" disponga de un coeficiente de garantía excepcional: el 14,2 %.

OBRA SOCIAL

Una gran inversión y un gran programa

Durante el año 1983 "la Caixa" destinó a la Obra Social más de 4.000 millones de pesetas para financiar todas sus actividades. Las acciones de la Obra Social durante este período han abarcado 312 localidades que se han beneficiado de un amplio programa de actividades y realizaciones de todo tipo.



Nuevos Centros de Recreo y Casas de Cultura

Este año han entrado en funcionamiento 15 nuevos equipamientos: 10 Centros de Recreo, 4 Casas de Cultura y 1 Complejo de Servicios Culturales. A finales de año, la Obra Social contaba con 263 equipamientos distribuidos en 164 poblaciones de Cataluña, Baleares, Madrid y Valencia.

41 Exposiciones de alto nivel

Los servicios de la Obra Social han organizado en 1983, 41 exposiciones que han contabilizado un total de 655.510 visitantes. También se ha celebrado en el Museo de la Ciencia, el Festival de Cine Científico y Cultural.

Intensa actividad educativa y cultural

El programa educativo «la Caixa a les Escoles» ha contado con la participación de más de medio millón de escolares. Por otra parte el Museo de la Ciencia es uno de los más visitados de España, con 246.995 visitantes.

La dotación de 34 Becas Fulbright, las publicaciones, las 107 Unidades de Experimentación Agrícola y la culminación de la colección «Las 100 mejores obras de la Literatura Catalana» son una buena muestra de la intensa actividad desarrollada.

Proyección territorial y ciudadana de la Obra Social

En conjunto, la actividad de la Obra Social de "la Caixa" se ha desarrollado en 312 municipios donde "la Caixa" también actúa como entidad financiera.

El programa de actividades ha totalizado 10.227 sesiones.

En total la Obra Social de "la Caixa" ha movilizado en 1983 más de 2.150.000 personas que han participado de una forma u otra en sus actividades.



```
0 REM***MUSICA MAESTRO C-64***
5 GOSUB 12000
10 CLR:REM NOTAS = VOZ 1
15 DIM HI(200),LO(200),LN$(200):REM
NUMERO TOTAL DE NOTAS POSIBLE = 200
20 POKE53281,7:POKE53280,7:PRINTCHR$
(144)CHR$(147)
30 UP$=CHR$(145):LF$=CHR$(157)
40 FORS=54272TO54296:POKES,0:NEXT
50 SV=54296:SH=54273:SL=54272:SA=542
77:SS=54278:SW=54276
100 PRINT"VOLUMEN[SPC][1-15]";CHR$(3
2)CHR$(32)"15"
101 PRINTUP$;:FORX=1TO14:PRINTCHR$(2
9);:NEXTX
102 INPUTV1$
105 V1=VAL(V1$)
110 IFV1<1ORV1>15THENGOSUB7000
114 IFV1<1ORV1>15THENPRINTUP$UP$UP$:
GOTO100
125 FORX=1TO22:PRINTCHR$(32);:NEXT:P
RINT
200 PRINT"FORMA[SPC]DE[SPC]ONDA"
202 PRINTTAB(10)CHR$(18)"T";CHR$(146
);"RIANGULAR"
204 PRINTTAB(10)CHR$(18)"S";CHR$(146
);"IERRA[SPC](DIENTE[SPC]DE-)"
206 PRINTTAB(10)CHR$(18)"P";CHR$(146
);"ULSADA"
208 PRINTTAB(10)CHR$(18)"N";CHR$(146
);"OISE[SPC](RUIDO)";CHR$(32)CHR$(32
)"T"
209 PRINTUP$;:FORX=1TO23:PRINTCHR$(2
9);:NEXT:INPUTW1$
210 IFW1$="T"THENW1=17:PRINT:GOTO300

212 IFW1$="S"THENW1=33:PRINT:GOTO300

214 IFW1$="P"THENW1=65:PRINT:GOTO300

216 IFW1$="N"THENW1=129:PRINT:GOTO30
0
220 IFW1$<>"N"THENGOSUB7000
226 IFW1$<>"N"THEN FORX=1TO6:PRINTUP
$;:NEXT:PRINT:GOTO200
300 PRINT"ATAQUE/DECAIMIENTO[SPC][0-
240]";CHR$(32)CHR$(32)"0"
302 PRINTUP$;:FORX=1TO26:PRINTCHR$(2
9);:NEXTX:INPUTA1$
310 A1=VAL(A1$)
320 IFA1<0ORA1>240THENGOSUB7000
330 IFA1<0ORA1>240THENPRINTUP$UP$:GO
TO300
400 PRINT"SUSTENTO/CAIDA[2SPC][0-240
]";CHR$(32)CHR$(32)"240"
402 PRINTUP$;:FORX=1TO23:PRINTCHR$(2
9);:NEXTX:INPUTS1$
410 S1=VAL(S1$)
420 IFS1<0ORS1>240THENGOSUB7000
430 IFS1<0ORS1>240THENPRINTUP$UP$:GO
TO400
450 PRINT:PRINT"DURACION[SPC]DE[SPC]
NOTA";CHR$(32)CHR$(32)"128"
451 PRINTUP$;:FORX=1TO16:PRINTCHR$(2
9);:NEXTX:INPUTD1$
452 IFLEFT$(D1$,1)<"0"ORLEFT$(D1$,1)
>"9"THENGOSUB7000
454 IFLEFT$(D1$,1)<"0"ORLEFT$(D1$,1)
>"9"THENPRINTUP$UP$UP$:GOTO450
456 D1=VAL(D1$):PRINT
458 FORX=1TO400:NEXTX
460 PRINTCHR$(147)CHR$(18)"F3"CHR$(1
46)"[SPC]TOCAR[SPC]Y[SPC]VER[SPC]";
461 PRINTTAB(20)CHR$(18)"F7"CHR$(146
)"[SPC]VOLVER[SPC]A[SPC]EMPEZAR"
462 PRINTCHR$(18)"[SPC]P"CHR$(146)"[SPC]
PAUSA";
463 PRINTSPC(12)CHR$(18)"RETURN"CHR$
(146)"[SPC]PIE[SPC]DE[SPC]LINEA"
464 PRINTCHR$(18)"[SPC]"CHR$(94)CHR$
(146)"[SPC]BORRAR[SPC]TODO";
465 PRINTSPC(6)CHR$(18)"INST/DEL"CHR
$(146)"[SPC]BORRAR[SPC]NOTA":PRINT:P
RINT
467 FORMD=0TO39:POKE1624+MD,67:POKE5
5896+MD,12:NEXTMD
480 GETA$:IFA$=""THEN480
499 REM **** A$ SELECCION DE NOTAS-
              OCTAVA 4
500 IFA$=CHR$(65)THEN HI=28:LO=49:GO
TO700:  REM  A
510 IFA$=CHR$(193)THEN HI=29:LO=223:
A$="A#":GOTO700:REM   A#
520 IFA$=CHR$(66)THEN HI=15:LO=210:G
OTO700: REM  B ** NOTA OCTAVA 3 NO 4
 ***
530 IFA$=CHR$(67)THEN HI=16:LO=195:G
OTO700: REM  C
540 IFA$=CHR$(195)THEN HI=17:LO=195:
A$="C#":GOTO700:REM   C#
550 IFA$=CHR$(68)THEN HI=18:LO=209:G
OTO700: REM  D
560 IFA$=CHR$(196)THEN HI=19:LO=239:
A$="D#":GOTO700:REM   D#
570 IFA$=CHR$(69)THEN HI=21:LO=31:GO
TO700:  REM  E
580 IFA$=CHR$(70)THEN HI=22:LO=96:GO
TO700: REM  F
590 IFA$=CHR$(198)THEN HI=23:LO=181:
A$="F#":GOTO700: REM   F#
600 IFA$=CHR$(71)THEN HI=25:LO=30:GO
TO700: REM  G
610 IFA$=CHR$(199)THEN HI=26:LO=156:
A$="G#":GOTO700: REM   G#
620 IFA$=CHR$(80)THEN HI=0:LO=0:A$="
P":GOTO700: REM  PAUSA
635 IFA$=CHR$(134)THEN 9000:REM F3 P
ARA  PONER LOS DATOS EN ARRAYS
640 IFA$=CHR$(136)THEN10:REM F7 VOLV
ER A EMPEZAR
645 IFA$=CHR$(13)THENPRINTUP$CHR$(13
):GOTO480:REM VOLVER A PIE DE LINEA
650 IFA$=CHR$(20)THENHI=0:LO=0:PRINT
CHR$(157)CHR$(157)CHR$(32);:GOTO480
651 REM LINE650 -INSERTAR/BORRAR UNA
 SOLA NOTA
660 IFA$=CHR$(94)THENGOTO460:REM LA
FLECHA ARRIBA BORRA TODA LA PANTALLA
690 IFA$<>CHR$(80)THENHI=0:LO=0:GOTO
480
700 POKESH,HI:POKESL,LO:POKESA,A1:PO
KESS,S1:POKESW,W1:POKESV,V1
740 FORX=1TOD1:NEXTX
750 PRINTA$;"[SPC]";:POKESV,16:POKES
W,W1-1
800 GOTO480
1999 REM * TOCAR UNA MELODIA *
2000 IFN=0 THEN RETURN
2005 FORX=54272 TO 54296:POKEX,0:NEX
TX
```

(Pasa a página 22)

casa de software s.a.

NOVEDADES:

- VIDEOCLUBS
- CONTAB. PERSONAL
- PERT
- ESTADISTICA

PRACTICALC 64



MANUAL EN CASTELLANO

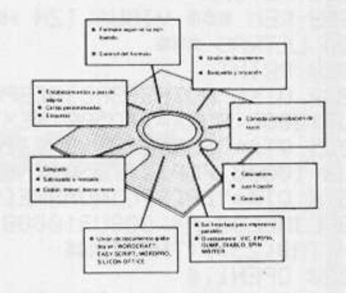
PROCESADOR DE TEXTO

35.000 caracteres, 240 columnas
Versión diskette: 21.500,–
Versión cartucho: 24.900,–
(grabación de documentos en diskette y cassette)

HOJA DE CALCULO

2.000 coordenadas
(funciones matemáticas, sort, gráficos...)
Versión cassette C-64: 15.500,–
Versión diskette C-64: 17.500,–
Versión cassette VIC-20: 11.500,–
Versión diskette VIC-20: 13.500,–

VIZAWRITE 64



MANUAL EN CASTELLANO

CONTABILIDAD 64 PROFESIONAL



VERSION: A	VERSION: B
300 cuentas	600 cuentas
3.000 apuntes	2.300 apuntes

- Listado de diario
- Balance de situación y de Sumas y Saldos
- Estractos de cuenta
- Cuenta de explotación, diario de cierre
- Mantenimiento de ficheros, utilitario...

EQUIPO NECESARIO:

- COMMODORE 64 y unidad de disco
- IMPRESORA MPS 801 (versión A y B)
- IMPRESORAS CENTRONICS (versión B)

P.V.P.: VERSION A: 24.550,–
VERSION B: 29.500,–

ADAPTADO AL PLAN GENERAL CONTABLE ESPAÑOL

VIDEO CLUBS

Programa en cartucho.
Necesario unidad de discos.

1.800 videofilms
650 clientes por disco
Saldo de cada cliente
Altas y bajas videofilms
Búsqueda videofilms/clientes.

P.V.P.: 35.000,–

LAPIZ OPTICO Y SOFTWARE GRAFICO



P.V.P. 12.500,–

PERT

Versión cassette

Para controlar el desarrollo de cualquier proyecto. Análisis de actividades, tiempos, nudos y caminos críticos.

Necesario impresora. P.V.P.: 4.000,–

VERSIONES VIC-20 y CBM 64

Dibujar en pantalla a mano alzada formas geométricas, sombreados...
Grabación del dibujo en cassette ó diskette.

CONTABILIDAD PERSONAL

26 conceptos
Listado por impresora y pantalla
Análisis financiero
Análisis porcentual gráfico

VERSION CASSETTE
P.V.P.: 6.000,–

JOYSTICK DE PRECISION

P.V.P. 2.950,–

ESTADISTICA

Versión cassette.

Cálculo del valor medio, desviación standard y coef. de variación.
Recta de regresión de valores de dos variables.

P.V.P.: 1.900,–

Pulsador sensible
Cable extra largo
Para CBM 64, VIC 20, ATARI CX 2600 y SPECTRUM y demás ordenadores con interface

CONTABILIDAD DOMESTICA

22 cuentas
Listado por pantalla o impresora de los ingresos y gastos de cada período definido.
Sin expansión de memoria

VERSION CASSETTE
P.V.P.:
C - 64: 3.500,–
VIC-20: 2.500,–

DE VENTA EN TIENDAS Y DISTRIBUIDORES ESPECIALIZADOS

casa de software, s.a.
c/. aragón, 272, 8.°, 6.ª
tel. 215 69 52
barcelona-7

DESEO RECIBIR INFORMACION:

Nombre y dirección:

COMODORE 64 ES UNA MARCA REG. DE COMMODORE BUSSINES MACHINES, INC.

(Viene de página 20)

```
2010 FORNN=0TON-1:POKESH,HI(NN):POKE
SL,LO(NN)
2020 POKESA,A1:POKESS,S1:POKESW,W1:P
OKESV,V1
2030 FORX=1TOD1:NEXTX
2040 NEXTNN:POKESV,16:POKESW,W1-1
2050 RETURN
2999 REM *** VISUALIZA NOTAS UTILIZA
NDO LETRAS ***
3000 PRINT
3002 QT$="NOTAS[SPC]A[SPC]LETRAS":GO
SUB10000:FORX=1TO500:NEXTX
3005 QT$="ENCIENDE[SPC]IMPRESORA":GO
SUB 10000:FORX=1TO500:NEXTX
3010 QT$="DA[SPC]NOMBRE[SPC]A[SPC]LA
[SPC]MELODIA":GOSUB10000:PRINTUP$:PR
INTTAB(23);:INPUTAW$
3020 OPEN1,4
3030 PRINT#1,AW$:PRINT#1
3040 FORNN=0TON-1:PRINT#1,LN$(NN);:P
RINT#1,"[SPC]";:NEXTNN
3050 PRINT#1:PRINT#1,"NUMERO[SPC]DE[SPC]
NOTAS[SPC]=";NN;"MAS[2SPC]-1,-1"
3090 PRINT#1:CLOSE1:GOTO9200
3095 PRINTUP$;:FORX=0TO39:PRINTCHR$(
32);:NEXTX:PRINTUP$UP$UP$:PRINT
3100 QT$="TERMINADA":GOSUB10000:GOTO
9200
7000 QT$="OPCION[SPC]ILEGAL":GOSUB10
000:FORX=1TO50:NEXTX
7010 PRINTUP$:FORX=1TO30:PRINTCHR$(3
2);:FORY=1TO25:NEXTY:NEXTX:RETURN
7999 REM *** SOLO IMPRIME LOS DATOS
**
8000 PRINT
8002 QT$="IMPRIME[SPC]SOLO[SPC]DATOS
":GOSUB10000:FORX=1TO500:NEXTX
8005 QT$="ENCIENDE[SPC]IMPRESORA":GO
SUB10000:FORX=1TO500:NEXTX
8010 QT$="DA[SPC]NOMBRE[SPC]A[SPC]LA
[SPC]MELODIA":GOSUB10000:PRINTUP$:PR
INTTAB(23);:INPUTAW$
8020 OPEN1,4
8030 PRINT#1,AW$:PRINT#1
8040 FORNN=0TON-1:PRINT#1,HI(NN);","
;LO(NN);",";:NEXTNN:PRINT#1,"-1,-1"
8045 PRINT#1:PRINT#1,"NUMERO[SPC]DE[SPC]
DATOS[SPC]=";NN*2;"MAS[SPC]-1,-1"
8050 PRINT#1:CLOSE1:RETURN
8499 REM *** IMPRIME TODO EL PROGRAM
A ***
8500 PRINT
8510 QT$="RUTINA[SPC]DE[SPC]IMPRESIO
N":GOSUB10000:FORX=1TO500:NEXTX
8520 QT$="ENCIENDE[SPC]IMPRESORA[SPC]
":GOSUB10000:FORX=1TO500:NEXTX
8530 QT$="DA[SPC]NOMBRE[SPC]A[SPC]LA
[SPC]MELODIA":GOSUB10000:PRINTUP$:PR
INTTAB(23);:INPUTAW$
8550 OPEN1,4
8560 PRINT#1,AW$
8565 PRINT#1
8570 PRINT#1,"100[SPC]FOR[SPC]S=5427
2[SPC]TO[SPC]54296[SPC]:[SPC]POKE[SPC]
S,0[SPC]:[SPC]NEXT[SPC]S"
8580 PRINT#1,"110[SPC]READ[SPC]SH,SL
"
8590 PRINT#1,"120[SPC]IF[SPC]SH=-1[SPC]
THEN[SPC]180"
8600 PRINT#1,"130[SPC]POKE54273,[SPC]
SH[SPC]:[SPC]POKE[SPC]54272,[SPC]SL"
8610 PRINT#1,"140[SPC]POKE[SPC]54277
,";A1;":[SPC]POKE[SPC]54278,";S1;":P
OKE[SPC]54276,";W1
8620 PRINT#1,"150[SPC]POKE[SPC]54296
,";V1
8630 PRINT#1,"160[SPC]FOR[SPC]X=1[SPC]
TO[SPC]";D1":[SPC]NEXT[SPC]X"
8640 PRINT#1,"170[SPC]GOTO[SPC]110"
8645 PRINT#1,"180[SPC]POKE[SPC]54296
,0[SPC]:[SPC]POKE[SPC]54276,";W1-1
8650 PRINT#1,"200[SPC]DATA[SPC]";
8660 FORNN=0TON-1:PRINT#1,HI(NN);","
;LO(NN);",";:NEXTNN:PRINT#1,"-1,-1"
8690 PRINT#1:PRINT#1,"NUMERO[SPC]DE[SPC]
DATOS[SPC]=";NN*2;"MAS[SPC]-1,-1"
8700 PRINT#1:CLOSE1:GOTO9200:REM SIG
UE CON OTRO A$ ***
8999 REM PONER DATOS EN ARRAYS
9000 PRINTCHR$(19);:FORX=0TO199:PRIN
TCHR$(32);:NEXTX
9005 PRINTCHR$(19);
9010 PRINTCHR$(18)"F1"CHR$(146)"[SPC]
SOLO[SPC]DATOS";
9011 PRINTTAB(17)CHR$(18)"F2"CHR$(14
6)"[SPC]IMPRIME[SPC]TODO"
9020 PRINTUP$;CHR$(18)"F3"CHR$(146)"
[SPC]PRACTICAR";
9022 PRINTSPC(5)CHR$(18)"F4"CHR$(146
)"[SPC]IMPRIME[SPC]NOTAS"
9025 PRINTCHR$(18)"F5"CHR$(146)"[SPC]
TOCA[SPC]MELODIA";
9030 PRINTTAB(17)CHR$(18)"F6"CHR$(14
6)"[SPC]DURACION"
9031 PRINTUP$;CHR$(18)"F7"CHR$(146)"
[SPC]VOLVER[SPC]A[SPC]EMPEZAR"
9032 PRINTCHR$(18)"[SPC]P"CHR$(146)"
[SPC]PAUSA";
9035 PRINTSPC(9)CHR$(18)"[SPC]"CHR$(
94)CHR$(146)"[SPC]BORRAR[SPC]TODO"
9040 PRINTCHR$(19):FORX=1TO11:PRINT:
NEXTX:REM COLOCAR EN LA LINEA GRIS
9100 PRINT"RUTINA[SPC]DE[SPC]INTRODU
CCION[SPC]EN[SPC]ARRAY";UP$
9150 A$=CHR$(94):GOTO9345:REM LEE PA
NTALLA Y FIJA N APORPIADAMENTE
9200 GETA$:IFA$=""THEN9200
9210 IFA$=CHR$(65)THEN HI(N)=28:LO(N
)=49:GOTO9400:REM  A
9220 IFA$=CHR$(193)THEN HI(N)=29:LO(
N)=223:A$="A#":GOTO9400:REM  A#
9230 IFA$=CHR$(66)THEN HI(N)=15:LO(N
)=210:GOTO9400:REM  B *NOTE OCTAVE 3
 NOT 4*
9240 IFA$=CHR$(67)THEN HI(N)=16:LO(N
)=195:GOTO9400:REM  C
9250 IFA$=CHR$(195)THEN HI(N)=17:LO(
N)=195:A$="C#":GOTO9400:REM  C#
9260 IFA$=CHR$(68)THEN HI(N)=18:LO(N
)=209:GOTO9400:REM  D
9270 IFA$=CHR$(196)THEN HI(N)=19:LO(
N)=239:A$="D#":GOTO9400:REM  D#
9280 IFA$=CHR$(69)THEN HI(N)=21:LO(N
)=31:GOTO9400:REM  E
9290 IFA$=CHR$(70)THEN HI(N)=22:LO(N
)=96:GOTO9400:REM  F
9300 IFA$=CHR$(198)THEN HI(N)=23:LO(
N)=181:A$="F#":GOTO9400:REM  F#
```

(Pasa a página 24)

(Viene de página 22)

```
9310 IFA$=CHR$(71)THEN HI(N)=25:LO(N
)=30:GOTO9400:REM  G
9320 IFA$=CHR$(199)THEN HI(N)=26:LO(
N)=156:A$="G#":GOTO9400:REM  G#
9330 IFA$=CHR$(80)THEN HI(N)=0:LO(N)
=0:GOTO9400:REM  P  PAUSA
9335 IFA$=CHR$(133)THENGOSUB8000:GOT
O9200:REM F1 IMPRIME SOLO LOS DATOS
9340 IFA$=CHR$(137)THENGOSUB8500:GOT
O9200:REM F2 RUTINA DE IMPRESION
9341 IFA$=CHR$(134)THENPRINTCHR$(147
):N=0:GOTO458:REM F3 PARA PRACTICAR
9342 IFA$=CHR$(135)ANDN=0THEN9200
9343 IFA$=CHR$(135)THENGOSUB2000:GOT
O9200:REM F5 TOCAR LA MELODIA
9344 IFA$=CHR$(136)THEN10:REM F7 COM
ENZAR TODO
9345 IFA$=CHR$(94)THEN10090
9346 IFA$=CHR$(138)THENGOTO3000:REM
F4 - IMPRIMIR NOTAS USANDO LETRAS
9347 IFA$=CHR$(139)THENPRINT:PRINT"D
URACION";:INPUT D1:GOTO9200:REM F6 D
URACION
9350 IFA$<>CHR$(80)THEN GOTO9200
9400 POKESH,HI(N):POKESL,LO(N):POKES
A,A1:POKESS,S1:POKESW,W1:POKESV,V1
9410 FORX=1TOD1:NEXTX
9420 PRINTA$;"[SPC]";:POKESV,16:POKE
SW,W1-1:LN$(N)=A$:REM FIJA LN - LIST
A NOTAS
9440 N=N+1:GOTO9200  :REM INCREMENTA
CUENTA Y LEE OTRA NOTA
9999 REM *** ESTA RUTINA HACE EL 'SC
ROLL'
10000 FORI=0TOLEN(QT$)
10010 POKESV,7:POKESH,126:POKESL,6:P
OKESA,120:POKESS,30:POKESW,33
10020 PRINTMID$(QT$,I+1,1);:POKESV,0
:POKESW,32:NEXTI
10030 PRINTUP$:FORX=1TO40:NEXTX
10040 FORS=54272TO54296:POKES,0:NEXT
S
10050 FORX=1TO100:NEXTX:RETURN
10089 REM *** BORRA ARRAY DE PANTALL
A ***
10090 ER=N+79:REM ER = BORRA NOTAS E
N ARRAY
10100 FORMD=0TOER:POKE1664+MD+1,62
10105 POKE55936+MD,1:POKE55936+MD+1,
2:NEXTMD
10110 PRINTCHR$(19):FORX=1TO12:PRINT
:NEXTX
10120 N=0:GOTO9200
11999 REM **TITULO DE PANTALLA**
12000 PRINTCHR$(147)CHR$(5):POKE5328
1,7:POKE53280,7
12002 FORS=54272TO54296:POKES,0:NEXT
12004 SV=54296:SH=54273:SL=54272:SA=
54277:SS=54278:SW=54276:UP$=CHR$(145
)
12010 FORX=1TO5:PRINT:NEXTX
12020 PRINT"MUSICA[SPC]MAESTRO":PRIN
T
12030 QT$="1984":GOSUB10000:PRINT:PR
INT
12040 QT$="BY[SPC]GARY[SPC]FIELDS":G
OSUB10000:FORX=1TO500:NEXTX
12050 PRINTUP$UP$UP$:QT$="VOZ[SPC]1[SPC]
/[SPC]OCTAVA[SPC]4":GOSUB10000:PRINT
:PRINT
12060 QT$="ESTO[SPC]CONVIERTE[SPC]EL
[SPC]SONIDO[SPC]EN[SPC]DATOS":GOSUB1
0000:PRINT:PRINT
12065 QT$="F1-[SPC]IMPRIME[SPC]SOLO[SPC]
DATOS":GOSUB10000:PRINT
12066 QT$="F2-[SPC]IMPRIME[SPC]PROGR
AMA[SPC]Y[SPC]DATOS":GOSUB10000:PRIN
T
12067 QT$="F4-[SPC]CONVIERTE[SPC]DAT
OS[SPC]EN[SPC]NOTAS":GOSUB10000:PRIN
T
12070 FORX=1TO500:NEXTX
12080 PRINTCHR$(147)
12090 PRINTCHR$(18)"[2SPC]LETRA[SPC]
=[SPC]NOTA[6SPC]":PRINT
12100 PRINT"[6SPC]A[SPC]=[SPC]A[3SPC]
"
12102 PRINT"[6SPC]B[SPC]=[SPC]B[3SPC]
"
12104 PRINT"[6SPC]C[SPC]=[SPC]C[3SPC]
"
12106 PRINT"SHIFT/C[SPC]=[SPC]C#[2SPC]
"
12108 PRINT"[6SPC]D[SPC]=[SPC]D[3SPC]
"
12110 PRINT"SHIFT/D[SPC]=[SPC]D#[2SPC]
"
12112 PRINT"[6SPC]E[SPC]=[SPC]E[3SPC]
"
12114 PRINT"[6SPC]F[SPC]=[SPC]F[3SPC]
"
12116 PRINT"SHIFT/F[SPC]=[SPC]F"  "
12118 PRINT"[6SPC]G[SPC]=[SPC]G[3SPC]
"
12120 PRINT"SHIFT/G[SPC]=[SPC]G#[2SPC]
"
12122 PRINT"[6SPC]P[SPC]=[SPC]PAUSA"
:PRINT
12124 PRINTCHR$(18)"OTRAS[SPC]TECLAS
[SPC]NADA":FORX=0TO39:PRINTCHR$(45);
:NEXTX
12125 PRINT"PUEDEN[SPC]MEMORIZARSE[SPC]
HASTA[SPC]200[SPC]NOTAS"
12128 QT$="LAS[SPC]NOTAS[SPC]PUEDEN[SPC]
SER":GOSUB10000:PRINT
12129 QT$="TOCADAS":GOSUB10000:FORX=
1TO100:NEXTX
12132 QT$="IMPRESAS":GOSUB10000:FORX
=1TO100:NEXTX
12134 QT$="LISTADAS[SPC]COMO[SPC]NOT
AS":GOSUB10000:FORX=1TO100:NEXTX
12136 QT$="IMPRESAS[SPC]COMO[SPC]PRO
GRAMA[SPC]CON[SPC]'DATAS'":GOSUB1000
0:PRINT
12200 PRINT"<UNA[SPC]TECLA>[SPC]PARA
[SPC]EMPEZAR":
12210 GET A$:IFA$=""THEN12210
12500 RETURN:REM ** VUELVE AL PROGRA
MA PRINCIPAL **
```

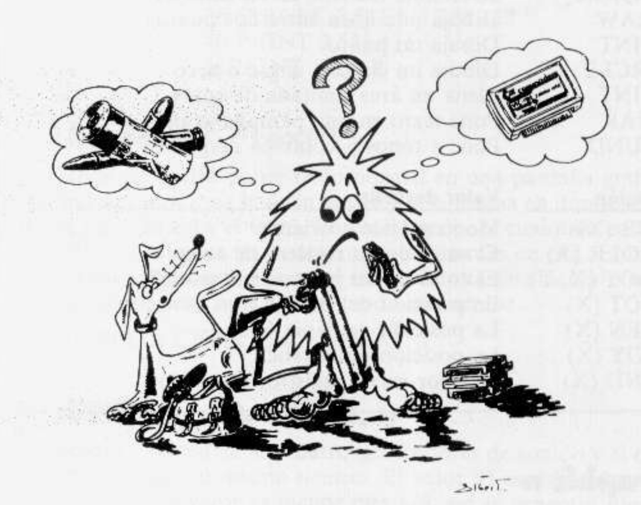


(Fin de programa)



PRIMERA PARTE

MANUAL DEL SUPER EXPANDER





VIC-20

1. INTRODUCCION

Su cartucho del Super Expander es el cartucho más versátil que usted puede comprar para su ordenador personal VIC-20. Añade un nuevo conjunto de color, de trazado de gráficos, de música y comandos de teclas de función al vocabulario de su VIC... y... añade 3K extras de memoria (RAM) por lo que usted puede escribir y ejecutar programas más largos. Con este cartucho puede trazar formas gráficas, "pintar" colores en la pantalla, redefinir fácilmente las teclas de función programables del VIC, escribir música... y más.

Los comandos y funciones que se dan en este manual funcionan como los comandos del BASIC. Por ejemplo, usted puede teclear el comando POINT1, 50, 50 y pulsar RETURN y el VIC colocaría un punto en esa posición de la pantalla.

Además, la mayoría de los comandos están preasignados a las teclas especiales de función de la derecha del teclado del VIC. Usted puede entrar esos comandos simplemente pulsando las teclas de función, para facilitar la programación con una tecla.

También vería que todos los comandos del BASIC funcionan normalmente con los comandos del Super-Expander. La única excepción es la sentencia IF...THEN. Toda declaración en Super Expander que vaya después de la palabra THEN debe separarse de dicha palabra por dos puntos (:) He aquí un ejemplo de cómo una sentencia se escribe correctamente y cómo no, usando la sentencia IF...THEN:

CORRECTO
```
IF X=THEN:SCNCLR
```

INCORRECTO
```
IF X=1 THEN SCNCLR
```

Los programas que han sido creados con las especiales sentencias del Super Expander no funcionarán en una máquina que no tenga el cartucho del Super Expander conectado.

Finalmente, la expansión extra de 3K de memoria se consigue automáticamente cuando se inserta el cartucho. En efecto, cuando encienda el VIC una vez conectado el cartucho, usted verá un incremento de aproximadamente 3000 bytes en la pantalla inicial.

Empezando el uso

1. Conecte su aparato de televisión.
2. Apague el VIC.
3. Inserte el cartucho.
4. Encienda el VIC.
5. El Super Expander está ahora activado y usted ya puede escribir programas en BASIC usando los comandos y sentencias especiales descritas en este manual.

2. Usando los gráficos del Super-Expander

Antes de empezar a trabajar con este manual es conveniente que le demos unas cuantas ideas aclaratorias. Para empezar, debe advertirse que, para tareas de trazados de gráficos, el cartucho del Super Expander divide el área de pantalla en 1024 por 1024 coordenadas. Por ejemplo, si usted teclea POINT1, 512, 512, verá aparecer un punto en la mitad de la pantalla. Esto son "pseudocoordenadas", lo que significa que el VIC acepta introducciones de datos basados en un sistema de coordenadas de puntos de 1024 por 1024 y hace que el trazado salga a escala de acuerdo con la actual relación de tamaño y resolución de la pantalla del VIC.

Una completa gama de instrucciones gráficas es automáticamente añadida a las instrucciones del BASIC estándard cuando inserta el cartucho del Super Expander. Estas instrucciones están explicadas a continuación.

NOTA: CUANDO TRABAJE CON GRAFICOS HAY DOS IMPORTANTES PARAMETROS QUE USTED DEBE INCLUIR AL PRINCIPIO DE SU PROGRAMA. Preparan la pantalla gráfica para que usted pueda trazar y dibujar, incluyen caracteres mezclados con gráficos, etc. Los dos comandos gráficos que usted debe poner al principio son: GRAPHIC y COLOR. El GRAPHIC coloca el VIC en el "MODO GRAPHIC" que usted quiera usar, y "COLOR" especifica los colores con los que quiere trabajar.

Instrucción	Efecto
GRAPHIC	Prepara la pantalla para gráficos
SCNCLR	Limpia una pantalla gráfica
COLOR	Selecciona la pantalla, el borde, el carácter y los colores auxiliares.



Instrucción	Efecto
REGION	Selecciona el color de un carácter
DRAW	Dibuja una línea entre dos puntos
POINT	Dibuja un punto
CIRCLE	Dibuja un círculo, elipse o arco
PAINT	Llena un área limitada de color
CHAR	Pone texto en una pantalla gráfica
SOUND	Pone 4 tonos y volumen de golpe

Función	Valor devuelto
RGR (X)	Modo gráfico corriente
RCOLR (X)	El valor en un registro de color
RDOT (X, Y)	El color de un punto de la pantalla
RPOT (X)	La posición del paddle
RPEN (X)	La posición del lápiz óptico
RJOY (X)	La posición del joystick
RSND (X)	El valor en un registro de sonido

Graphic n

Prepara la pantalla gráfica y activa el conjunto de instrucciones gráficas. Hay 5 modos GRAPHIC diferentes, los cuales son activando tecleando GRAPHIC y el número, como instrucción en su programa.

Ejemplo:

```
1Ø GRAPHIC 2
2Ø COLOR 1,2,3,4
```

Teclee RUN y pulse RETURN.

Valor de n	Modo
Ø	Modo texto (normal)
1	Modo multicolor
2	Modo de alta resolución
3	Modo de vuelta al texto

El modo Ø es el modo normal en el VIC. Una vez este modo ha sido cambiado, la sentencia GRAPHIC 4 es usado para volver a este modo.

En el modo 1 hay 8Ø puntos por línea, y 16Ø líneas. Cada punto está dibujado en uno de los cuatro registros de color: color de pantalla, color del borde, color de carácter o color auxiliar. Un punto colocador en el color de pantalla es invisible, ya que tiene el mismo color que el fondo de la pantalla; esto puede ser usado para borrar otros puntos. Los puntos dibujados en cualquiera de los colores de pantalla, de borde o auxiliares cambiarán automáticamente cuando esos colores cambien. El carácter de color está asociado con cada específico espacio de 4 por 16 puntos de la pantalla. Esto significa que si un punto es colocado en ese espacio y en un carácter de color, todos los otros puntos de ese espacio que estuvieran en el carácter de color cambiarán al nuevo color.

En el modo 2, hay 16Ø puntos por línea, y 16Ø líneas. Los puntos tienen la mitad de anchura que en el modo 1, lo cual consigue una resolución dos veces mejor en la pantalla. No obstante, la resolución del color es más limitada. Los puntos pueden ser trazados sólo en un carácter de color. Dentro de cada específico espacio en la pantalla, que es un área de 8 por 8 puntos, sólo puede haber un carácter de color, por lo que usted sólo puede tener puntos de ese color en la pantalla.

El modo 3 es una mezcla de los modos 2 y 1. Cuando el color de carácter es menor que 8, el trazado será hecho con alta resolución. Cuando el color de carácter es mayor que 7, obtenemos el modo multicolor.

El modo 4 vuelve la pantalla al modo de texto.

ATENCION: Si el VIC ya está en modo texto, aparecerán cosas raras en la pantalla y usted tendrá que apagarlo y volverlo a encender.

SCNCLR

Esta sentencia limpia la pantalla que está en modo gráfico. Puede ser tecleado directamente o incluida en una línea de un programa.

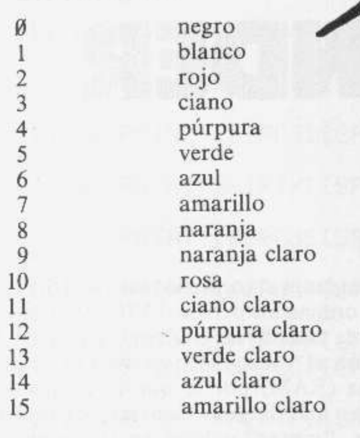
COLOR sc, bo, ch, au

Esto inicializa y prepara los cuatro diferentes registros de color. Las variables están limitadas por:

sc = Ø a 15
bo = Ø a 7
ch = Ø a 7 normal, 8 a 15 multicolor
au = Ø a 15

Los colores son:

Ø	negro
1	blanco
2	rojo
3	ciano
4	púrpura
5	verde
6	azul
7	amarillo
8	naranja
9	naranja claro
10	rosa
11	ciano claro
12	púrpura claro
13	verde claro
14	azul claro
15	amarillo claro





NOTA: en el modo GRAPHIC 3, los colores de carácter mayores que 7 colocan el modo multicolor. En ese caso, el actual carácter en ese espacio es el color número menos 8.
EJEMPLO: COLOR 1, 2, 3, 4

REGION c

Esto funciona como la sentencia COLOR, con la diferencia de que sólo el color de carácter es afectado.

EJEMPLO: REGION 7

DRAW c, x1, y1 TO x2, y2 [TO x3, y3...to xn, yn] o DRAW c TO x1, y1 [...]

Esto le permite dibujar líneas en la pantalla. La variable c es el color de la línea, según el siguiente esquema:

c	alta resolución	multicolor
Ø	pantalla	pantalla
1	carácter	borde
2	carácter	carácter
3	carácter	auxiliar

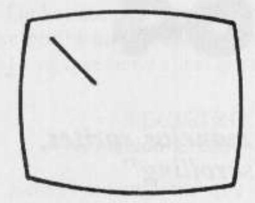
La x y la y son las variables de las coordenadas de los puntos que indican el final de la línea. Los valores de la X y la Y pueden estar comprendidos entre Ø y 1Ø23. El valor de X Ø indica el margen izquierdo de la pantalla, y X=1Ø23 indica el margen derecho. El valor de la Y sitúa la línea arriba y abajo. Los valores de Ø a 1Ø23 están a la escala del tamaño de la pantalla (16Ø por 16Ø ó 8Ø por 16Ø).



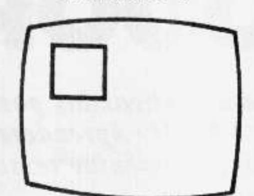
EJEMPLO 1: DRAW 1, Ø, Ø TO 1ØØ, 1ØØ

EJEMPLO 2: DRAW 1,Ø,Ø TO Ø, 1ØØ TO ¡ØØ, 1ØØ, TO 1ØØ, Ø TOØ, Ø

EJEMPLO 1



EJEMPLO 2



POINT c, x, y
o
POINT c, x1, y1, x2 y 2..., xn, yn

Esto coloca un punto o puntos en la pantalla con el color especificado. Los puntos están dados en la forma de Ø a 1Ø23.

EJEMPLO 1: POINT 1, 5ØØ, 5ØØ

EJEMPLO 2: POINT 2, 5ØØ, 5ØØ, 6ØØ, 6ØØ, 7ØØ, 7ØØ

CIRCLE c, x, y, rx, ry +, as, ae

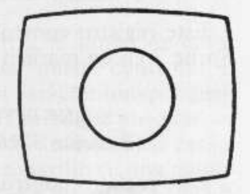
c es el registro de color (vea la tabla en la sección DRAW) en el cual se dibuja el círculo. x e y son las coordenadas del centro, rx y ry son la anchura y la altura del círculo. as y ae son los ángulos opcionales que marcan la longitud del arco. Van en gradianes (Ø a 1ØØ grados en un círculo completo.

EJEMPLO 1: CIRCLE 1, 511, 511, 3ØØ, 4ØØ

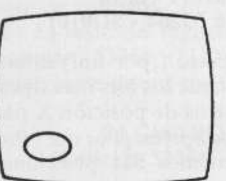
EJEMPLO 2: CIRCLE 1, 2ØØ, 9ØØ, 1ØØ, 1ØØ

EJEMPLO 3: CIRCLE 2, 5ØØ, 6ØØ, 2ØØ, 3ØØ, 1Ø, 7Ø

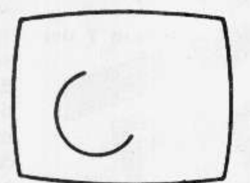
EJEMPLO 1



EJEMPLO 2



EJEMPLO 3



PAINT c, x, y

Estas pinturas están encerradas en un área encerrada en un color. La c es el registro de color, y la x y son cualquier punto dentro del área en ser pintada. Cada área puede ser pintada una sola vez. A causa de las limitaciones de color de algunos modos GRAPHIC, algunas formas pueden ser distorsionadas en esta operación. Si una área está limitada por áreas en modo multicolor; debe ser pintada en modo multicolor.

EJEMPLO

```
1Ø GRAPHIC 1
2Ø COLOR 1,7,Ø,1Ø
3Ø CIRCLE 2,511,511,2ØØ,3ØØ
4Ø PAINT 3,511,511
5ØPAINT 1,511,Ø
```

CHAR ro, co, "text"

Esto le permitirá poner texto normal en una pantalla gráfica. Las variables ro y co indican la fila y la columna en donde usted desea que aparezca el texto. El texto puede ser cualquier cadena variable o cualquier texto dentro de las marcas de cita. El texto aparecerá sólo si estamos en el modo de alta resolución o en el modo mixto, pero nunca en el modo multicolor.

EJEMPLO: CHAR 1, 1, "HOLA"

SOUND s1, s2, s3, s4, v

Prepara e inicializa los cuatro generadores de sonido y el control de volumen al mismo tiempo. El valor S1 es colocado en el registro 1. Si el valor es menor que 128, no se generará ningún sonido. El volumen puede variar de Ø(no suena nada) hasta 15. Esto le permite crear y generar gran cantidad de sonidos.

EJEMPLO: SOUND 225, 225, 225, Ø, 15

RGR (x)

Esta funciñn es el modo gráfico que fue introducido por el comando GRAPHIC. El valor de x va de Ø a 255, y no afecta el resultado.

EJEMPLO: X=RGR(Ø)

RCOLR (x)

Lee el color de cualquiera de los cuatro registros, según sea el valor de x, atendiendo a la siguiente tabla:

x	Registro de color
Ø	color de pantalla
1	color de borde
2	color de carácter
3	color auxiliar

EJEMPLO: X=RCOLR(2)

RDOT(x, y)

Lee el valor del color de un punto en la pantalla. Los valores de x y de y son las coordenadas de dicho punto, y el valor del color devuelto es el registro de color de ese espacio.

EJEMPLO: X= RDOT(511, 511)

RPOT(x)

Lee el valor del paddle. Si x=Ø, la X del paddle es leída, y si x=1, se lee la Y del paddle.

EJEMPLO:

```
1ØGRAPHIC2:COLOR11,6,6,6
2Ø X=RPOT(Ø):Y=RPOT(1):J=RJOY(Ø)
3Ø IFJ=12THEN:SCNCLR
4Ø IFJ=4THEN:REGION2
5Ø IFJ=8THEN:REGION6
6Ø POINT2,X*4,Y*4:GOTO20
```

Finalizará en el próximo número



EL CHIP DE VIDEO DEL COMMODORE-64

En esta serie de artículos intentaré explicar las posibilidades del chip de vídeo del COMMODORE 64, 6569 ó VIC-II. Aprenderemos cómo manejar sprites, cómo crear pantallas de Alta Resolución, cómo presentar textos con un "scrolling" punto a punto, etc.

Por R. PARDO

Vamos a ver los registros del VIC-II

¿Qué entendemos por registro? Bien, un registro es un grupo de varios bits (incluso de bytes) que cumplen una función específica en un chip de control de periférico. De hecho, el procesador central "ve" a estos registros como posiciones normales de memoria, y para el chip de control de periférico esta misma posición de memoria tiene una tarea específica. Por ejemplo: los registros, en el caso del chip VIC-II controlan el acceso a cualquiera de las funciones que este chip pueda soportar. De hecho, actuando sobre un registro X le diremos al chip de vídeo dónde tiene que situar horizontalmente el Sprite número 0 y actuando sobre el registro X + 1 le diremos la posición vertical del mismo Sprite...

Habiendo quedado claro el concepto de registro, revisaremos en primer lugar todos los registros de este chip.

REGISTRO 0
(Posición 53248 - $D000)

Este registro controla la posición X del Sprite 0 en un margen de 0 a 225.

REGISTRO 1
(Posición 53249 - $D001)

Este registro controla la posición Y del Sprite 0 en un margen de 0 a 255.

REGISTRO 2
(Posición 53250 - $D002)

Este registro controla la posición X del Sprite 1 en un margen de 0 a 255.

REGISTRO 3
(Posición 53251 - $D003)

Este registro controla la posición Y del Sprite 1 en un margen de 0 a 255.

REGISTRO 4
(Posición 53252 - $D004)

Este registro controla la posición X del Sprite 2 en un margen de 0 a 255.



REGISTRO 5
(Posición 53253 - $D005)

Este registro controla la posición Y del Sprite 2 en un margen de 0 a 255.

REGISTRO 6
(Posición 53254 - $D006)

Este registro controla la posición X del Sprite 3 en un margen de 0 a 255.

REGISTRO 7
(Posición 53255 - $D007)

Este registro controla la posición Y del Sprite 3 en un margen de 0 a 255.

REGISTRO 8
(Posición 53256 - $D008)

Este registro controla la posición X del Sprite 4 en un margen de 0 a 255.

REGISTRO 9
(Posición 53257 - $D009)

Este registro controla la posición Y del Sprite 4 en un margen de 0 a 255.

REGISTRO 10
(Posición 53258 - $D00A)

Este registro controla la posición X del Sprite 5 en un margen de 0 a 255.

REGISTRO 11
(Posición 53259 - $D00B)

Este registro controla la posición Y del Sprite 5 en un margen de 0 a 255.

REGISTRO 12
(Posición 53260 - $D00C)

Este registro controla la posición X del Sprite 6 en un margen de 0 a 255.

REGISTRO 13
(Posición 53261 - $D00D)

Este registro controla la posición Y del Sprite 6 en un margen de 0 a 255.

REGISTRO 14
(Posición 53262 - $D00E)

Este registro controla la posición X del Sprite 7 en un margen de 0 a 255.

REGISTRO 15
(Posición 53263 - $D00F)

Este registro controla la posición X del Sprite 7 en un margen de 0 a 255.

REGISTRO 16
(Posición 53264 - $D010)

Este es un registro (¡por fin!) diferente. Este registro contiene los bits más significativos de los registros de posición X para el movimiento de los Sprites ¿Por qué? Porque con un sprite tenemos **511** posiciones de pantalla para movernos (parte del movimiento del sprite es invisible con respecto a nosotros ya que transcurren detrás del marco) y nosotros hasta ahora y con los registros de movimiento horizontal sólo podíamos movernos por la parte izquierda de la pantalla. Luego nosotros debemos prever siempre en nuestros programas que usen sprites la posibilidad de que se tengan que mover más allá del límite 255 y en ese momento pondremos el bit correspondiente de este registro a 1.

Ahora veremos cómo están repartidos:



7 6 5 4 3 2 1 0
0 0 0 1 0 0 0 0

Donde el bit más significativo del Sprite 7 será el bit 7 y el bit más significativo del sprite 0 sera el bit 0.

De hecho, en el ejemplo que doy, el sprite 4 se podrá mover en el área de pantalla 256 - 511 ya que tiene el bit correspondiente a 1.

REGISTRO 17
(Posición 53265 - $D011)

Este es un registro que controla algunos de los modos de trabajo del chip de vídeo. Ahora veremos, a nivel de bit, lo que ejecuta este registro:

Bits 0, 1 y 2: estos bits contienen la posición vertical (de 0 a 7) de la línea en la cual queremos ejecutar "smooth scrolling" (scrolling suave, punto a punto).

Bit 3: este bit selecciona el número de líneas visibles en pantalla. Cuando este bit está a 0, la pantalla pasa a tener 24 líneas. Cuando está a 1, la pantalla dispone de 25 líneas visibles. Esta característica se usa para técnicas especiales de presentación.

Bit 4: este bit activa el "blanking" de pantalla (el fondo pasa a ser del mismo color del marco, sin borrar ninguna información de la pantalla) y, de paso, desactiva parte de las actividades del chip de vídeo con lo cual permite que el microprocesador tenga un acceso sin restricciones al bus. Cuando este bit está a 0 activa el "blanking".

Bit 5: este bit activa el modo "bit-map" que es, en definitiva, el que permite el trazado de funciones en Alta Resolución. Cuando este bit está a 1, se activa el modo "bit-map".

Bit 6: este bit activa el modo de color extendido, en el cual puede controlar el color de fondo de un carácter independientemente del color de fondo que puede tener cualquier otro carácter (mostrar un carácter azul sobre fondo amarillo en una pantalla blanca). Cuando este bit está a 1, queda activado el modo de color extendido.

Bit 7: éste es el bit más significativo del registro de comparación "raster" (barrido) (bit 8). El resto del registro está situado en la posición 53266 ($D012). Hablaremos detenidamente de este registro más adelante.

REGISTRO 18
(Posición 53266 - $D012)

En esta posición tenemos los 8 bits menos significativos del registro de comparación raster. Si leemos este registro, obtendremos los bits más bajos de la posición actual del barrido en la pantalla (recuerde que el bit más significativo de este registro se encuentra en la posición 53265 - $D011). Se puede usar este registro para que los cambios en la pantalla se introduzcan sin perturbaciones en la misma. Esto se consigue cuando el barrido no está en las posiciones visibles, es decir, entre las posiciones 51 y 251. Cuando se escribe en este registro, el valor asignado se guarda para usarlo en la comparación con el valor actual de la posición del barrido en la pantalla. Si estos dos valores son iguales, el bit 0 del registro de interrupción del chip de vídeo se colocará a 1.

REGISTRO 19
(Posición 53267 - $D013)

Este registro contiene los 8 bits más significativos de la posición horizontal del lápiz óptico en la pantalla. Ya que la posición horizontal del lápiz óptico está definida por un contador de 512 posiciones (9 bits), se obtendrá una resolución máxima de 2 puntos horizontales.

REGISTRO 20
(Posición 53268 - $D014)

Este registro contiene la posición vertical del lápiz óptico, pero al tener sólo 8 bits se obtiene el valor de barrido dentro de la pantalla.

REGISTRO 21
(Posición 53269 - $D015)

Este es el registro de activación de Sprites. El bit 0 activará al Sprite 0, el bit 1 activará al Sprite 1..., y el bit 7 activará al Sprite 7, entendiendo como activar el hecho de que sea visible o no el Sprite seleccionado poniendo a 1 ó no el bit seleccionado (1=activado).

REGISTRO 20
(Posición 53270 - $D016)

Este registro cumple algunas funciones importantes del chip de vídeo. Veamos esta posición bit a bit:

Bits 7 y 6: sin uso, normalmente están a 1.

Bit 5: ¡mantenga siempre este bit a 0! ¿Por qué? Este bit controla la secuencia de reset del chip de vídeo. Cuando se inicialice el chip de vídeo, este bit debe estar a 0. Cuando se pone a 1, todas las operaciones del chip de vídeo se interumpen, incluyendo salida de vídeo y sincronismos, refresco de memoria RAM, y el acceso al bus del sistema.

Bit 4: este es el bit que selecciona el Modo Multicolor, tanto en Modo Texto como en Modo Bit-Map. Se activa cuando está a 1.

Bit 3: este bit selecciones el número de columnas disponibles en la pantalla, que puede ser de 38 ó 40 columnas. Cuando esté bit a 1, está en Modo 40 columnas. Este tipo de características se usa para técnicas de presentación especial.

Bits 2, 1 y 0: estos bits seleccionan la posición horizontal de la línea (de 0 a 7) en la cual queremos ejecutar Modo Smooth Scrolling (Scrolling Suave). ■



Rincón del 700

SOFTWARE PARA EL 700 (7) UTILITARIOS (4)

DISKMON

Por Jordi SASTRE

Este mes vamos a ver un utilitario muy interesante, el DISKMON. Su cometido es visualizar y/o modificar el contenido de bloques de disco.

En un momento dado puede ser necesario acceder a un bloque de disco para analizar su contenido o modificar algunos bytes. En Basic puede hacerse perfectamente con pequeños programas (recordemos el programa que vimos en el número 3 de esta revista —Pág. 40— para visualizar el contenido de un bloque):

```
10 OPEN 15,8,15 : OPEN 2,8,2,"#"
20 PRINT#15,"U1";2;0;T;S:REM T=
   TRACK S=SECTOR
30 FOR X=1 TO 255
40 GET#2, A$ : PRINT A$;
50 NEXT
60 CLOSE 2 : CLOSE 15
```

Si en lugar del carácter ASCII, queremos visualizar el valor absoluto de cada byte, debemos modificar la línea 40 de la siguiente manera:

```
40 GET#2, A$ : PRINT ASC
   (A$+CHR$(0));
```

Nótese que el valor ASCII se obtiene añadiendo a la cadena un CHR$(0) al final. Ello es debido a que cuando se lee un byte cuyo valor es cero, la variable de recepción (en este caso A$) es asignada como cadena nula. Si se intenta obtener el ASCII de una cadena nula: ASC (""), el ordenador emite un mensaje de error "?ILLEGAL QUANTITY". Ello no sucede si se le añade el CHR$(0) al final. El CHR$(0) no afecta al valor ASCII de la cadena cuando esta no es nula.

Continuando con nuestros ejemplos, si sólo se desean visualizar 10 bytes a partir del byte 50, debemos posicionar el puntero y cambiar el valor del bucle FOR/NEXT:

```
25 PRINT#15,"B-P";2;50
30 FOR X=1 TO 10
```

Por otro lado, si lo que queremos es poner a cero el byte 10 del track 5, sector 20, debemos ejecutar el siguiente programa:

```
10 OPEN 15,8,15 : OPEN 2,8,2,"#"
20 PRINT#15,"U1";2;0;5;20
30 PRINT#15,"B-P";2;10
40 PRINT#2,CHR$(0);
50 PRINT#15,"U2";2;0;5;20
60 CLOSE 2 : CLOSE 15
```

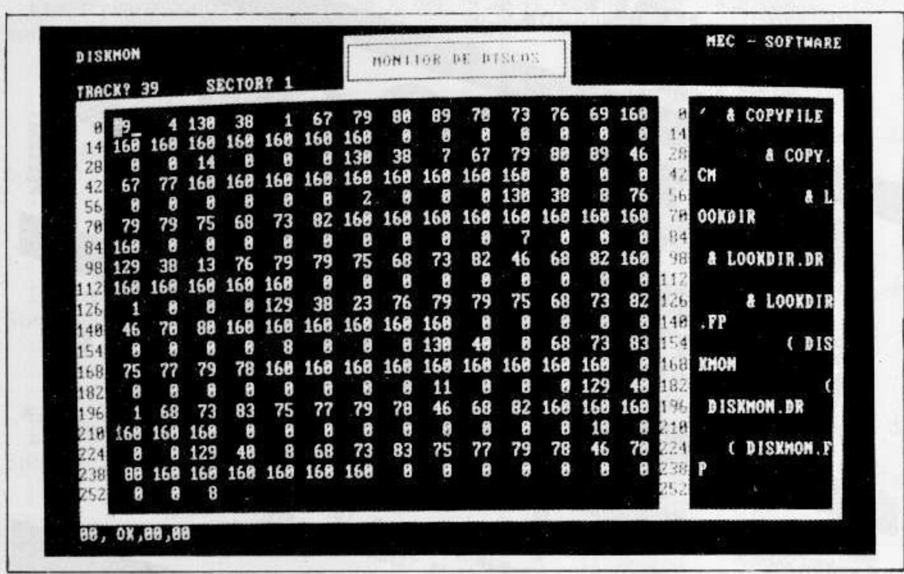
Como vemos, variantes de un mismo programa permiten realizar múltiples y diversas funciones. Cuando queremos "arreglar" algún fichero o modificar cualquier cosa del disco, es difícil hacer un programa que automáticamente lea unos bytes, los analice, tome las decisiones pertinentes y haga las modificaciones oportunas. Normalmente lo que se hace es, partiendo de un programa base (que puede ser cualquiera de los que hemos indicado hasta ahora), efectuamos todas las consultas, lecturas y grabaciones de bytes que sean necesarias de una manera manual, de una en una, modificando el programa a medida que sea necesario.

Este sistema es muy artesanal y meritorio para el programador, pero encierra grandes riesgos que debe intentarse evitar (una equivocación en el número de sector al hacer la grabación puede destruir irreparablemente información vital). Menos peligroso es cuando la información se manipula bajo un programa que se encarga de controlar que los bloques se graban siempre en el mismo track y sector de donde fueron leídos, que identifique claramente cada byte con su número, que no sea complicado de manejar y así evitar errores del operador, que visualice a la vez los valores ASCII y absoluto de cada byte,... etc. Qué casualidad!, hay un programa, llamado DISKMON, que hace todo esto y más.

Lógicamente, para manejar la información directamente sobre el disco, debe conocerse muy bien cómo están guardados los datos: estructura del disco, BAM, encadenamiento entre bloques de ficheros, etc. Por tanto, antes de pasar a comentar el utilitario en sí, conviene dar un repaso a estos conceptos.

Es posible que esta descripción la empiece en un nivel muy bajo para los que ya han manejado discos alguna vez, pero siempre conviene hacer caso de una de las sabias normas de todo programador: dar algo por supuesto es la manera más rápida de tener problemas.

Para empezar debemos saber que un disco está dividido en TRACKS, y cada track está dividido en SECTORES. A un sector concreto de un track lo llamamos BLOQUE. Todos los bloques tienen 256



"Diskmon en acción (diseño: J. Sastre; Foto: P. Masats").



bytes cada uno, numerados del 0 al 255. El número de tracks varía según la unidad de discos de que se trate (el 8050 tiene 77 tracks, el 8250 tiene 154, etc.), y el número de sectores varía según el track tratado (en un track ubicado en la parte "exterior" del diskette caben más sectores que en uno de la zona "interior", porque el primero es más largo). En la Tabla 1 se detalla la estructura de disco de la unidad 8250, que es la que voy a utilizar como base y ejemplo en este artículo. El 8250 no se diferencia excesivamente de otras unidades de disco (3040, 8050, 9060,... etc.) en cuanto a sistemas operativos, pero sí en cuanto a estructura de tracks y sectores. Por ejemplo, el directorio del disco está ubicado a partir del track 39 en el 8050 y 8250, mientras que en el 2031, 3040 y 4040 está a partir del track 18.

En adelante, cuando haga referencia a un bloque, lo identificaré con sus números de track y sector separados por una coma, por ejemplo: 20,8 se refiere al bloque situado en el track 20, sector 8.

De la Tabla 1 vemos en primer lugar el reparto exacto de tracks y sectores que tiene el disco; sabemos, por ejemplo, que el bloque 20,82 o 241,15 no existe nunca. También notamos que los tracks se numeran a partir del 1, mientras que los sectores empiezan en el 0. Por último, sabemos que todos los bloques del track 39 y los bloques 39,0, 38,3, 38,6 y 38,9 están reservados por el sistema para guardar la cabecera de directorio, el directorio y el BAM. Esto nos provoca la diferencia entre el número total de bloques existentes en el disco y el número de bloques que realmente está disponible para el usuario. Pero ¿qué es la cabecera de directorio, el directorio y el BAM?

La CABECERA DE DIRECTORIO (en inglés "Directory Header") es el bloque donde el sistema guarda datos referentes al disco en conjunto: nombre del disco, identificador,... etc., tal como se indica en la Tabla 2.

El DIRECTORIO son una serie de bloques donde el sistema guarda información acerca de cada una de las entradas en directorio (programas y ficheros). El comando CATALOG o DIRECTORY visualiza parte de esta información (ver Tabla 3).

El BAM (del inglés "Block Availability Map" = Mapa de Bloques Disponibles) es un bitmap por el cual el sistema sabe qué bloques están ocupados y cuáles no.

Si queremos acceder a un fichero, necesitamos saber también cómo están relacionados entre sí los diferentes bloques que componen el fichero en su totalidad (los programas están incluidos en el término "ficheros").

Los ficheros se componen de una serie de bloques físicamente no consecutivos. El Directorio indica cuál es el primero de ellos, mientras que cada uno apunta al siguiente sucesivamente hasta el último bloque. Por tanto, la estructura de un bloque de datos (de programa o de fichero secuencial o relativo) es:

BYTES	CONTENIDO
0-1	Bloque (track, sector) del siguiente bloque.
2-255	254 bytes de información

Cuando el track es cero (byte cero = 0), significa que éste es el último bloque del fichero. Además, si se trata de un programa, el final del mismo viene indicado dentro del bloque por tres ceros consecutivos.

En los ficheros relativos raramente coincide el final del registro lógico con el final del bloque físico. Por tanto, frecuentemente nos encontraremos con que un bloque de disco contiene parte de un registro, uno o más registros completos, y el inicio del siguiente registro, cuyo final está en el siguiente bloque.

He hecho una descripción muy rápida sobre los discos. Todo lo dicho está también explicado en el Manual de las Unidades de Disco Commodore. Por tanto, si alguien desea profundizar sus conocimientos sobre este tema puede remitirse a este Manual.

Funcionamiento del programa

Pasaremos ahora a describir el manejo del utilitario DISKMON en concreto. DISKMON se compone de tres ficheros:

DISKMON.—Programa Basic.

DISKMON.DR.—Descriptor de pantalla asociada.

DISKMON.FP.—Pantalla asociada.

Esto significa que si en un momento dado se desea copiar el utilitario DISKMON de un disco a otro, deben copiarse estos tres ficheros.

DISKMON se pone en marcha llamando y ejecutando el programa DISKMON. Si el formato de pantalla (DISKMON.DR y DISKMON.FP) no estuviera contenido en el drive 0 del periférico 8 (que es donde el programa lo busca), deberá cambiarse la línea 1010 del programa DISKMON.

La pantalla se carga y visualiza inmediatamente. Es en esta pantalla donde se visualizarán los contenidos de los bloques, a la izquierda el valor decimal de cada byte y a la derecha su equivalente en ASCII.

En la línea 25 de la pantalla el operador debe introducir el número de periférico y el drive a cuyos bloques de disco desea acceder.

En la parte superior el programa pide TRACK y SECTOR del bloque a visualizar. Si se introducen valores correctos el programa carga el bloque y visualiza su ▶



contenido en la pantalla. En la última línea informa del status (DS$) obtenido al cargar el bloque, por si hubiera encontrado algún error.

Mientras se visualizan los bytes, puede pulsarse ESC para interrumpir el proceso, yendo el programa a pedir un nuevo TRACK y SECTOR.

Una vez visualizado todo el bloque, el cursor se posiciona en el byte 0, permitiendo su modificación. La tecla RETURN se desplaza al byte siguiente, SHIFT/RETURN al byte anterior y ESC finaliza la modificación.

El programa pide conformidad antes de grabar de nuevo el bloque en el disco. Después vuelve a pedir otro TRACK y SECTOR. Pulsando RETURN, SHIFT/RETURN o ESC sin introducir nada en TRACK, el programa finaliza.

Aplicaciones

Alquien puede pensar que el acceso "fraudulento" sobre el disco que éste utilitario ofrece es innecesario, bien porque mediante programas Basic pueden conseguirse resultados parecidos, o bien porque no se le aprecia a primera vista ninguna gran ventaja que lo justifique.

El DISKMON sirve para modificar bytes del disco. De acuerdo que estos bytes pueden modificarse por otros medios si forman parte de ficheros accesibles vía Basic. Pero la utilidad del DISKMON está principalmente en dos aspectos: primero porque trabaja sobre bloques físicos de disco en lugar de records de fichero relativo (asi es el caso), y el segundo porque permite acceder a bytes que normalmente no son accesibles: podemos visualizar el BAM, el directorio, los bloques que componen los programas, etc.

Por ejemplo. En la Tabla 3 se indica que el byte 2 corresponde al Tipo de Fichero. También se dice que el tipo de fichero 0 es "fichero borrado" (?!). Cuando un fichero se borra del disco (SCRATCH), el sistema libera los bloques que ocupa y pone el byte de Tipo de Fichero a cero. Por tanto, si después de borrar un fichero restauramos este byte a su valor anterior (según el tipo que era), podremos operar de nuevo con el fichero: lo hemos resucitado!. Después hemos de volver a reservar en el BAM los bloques que ocupaba, o bien, si se trata de un programa, una vez cargado en memoria podemos grabarlo con otro nombre antes de volver a poner el byte de Tipo de Fichero a cero para no complicarnos la vida con el BAM.

Otra posible utilidad del DISKMON es cuando un fichero da errores de lectura. Entonces podemos buscar el bloque erróneo y sustituirlo por otro modificando el encadenamiento entre bloques (si es relativo debe modificarse también el Side-Sector).

Con el DISKMON podemos cambiar el nombre de un disco, el identificador, nombres de ficheros, etc.; pero lo más importante es que con él podemos "reparar" ficheros "estropeados". ■

Tabla 1: Mapa de Distribución de Bloques (8250)

TRACKS	SECTORES	SECTORES POR TRACK	
1-39	0-28	29	Total bloques: 4,166
40-53	0-26	27	Track de directorio: 39
56-64	0-24	25	Bloque de cabecera: 39,0
65-77	0-22	23	Bloques de BAM: 38,0, 38,3, 38,6 y 38,9
78-116	0-28	29	Bloques disponibles: 4. 133
117-130	0-26	27	
131-141	0-24	25	
142-154	0-22	23	

Tabla 2: Cabecera de Directorio (Track 39 Sector 0) (8250)

BYTES	VALOR	CONTENIDO
0-1	38,0	Track y sector del primer bloque de BAM.
2	67	Identifica formato DOS 2,7
3-5		No usados
6-21		Nombre del disco, rellenado con CHR$(160).
22-23	160	Constantes.
24-25		Identificador del disco.
26	160	Constante.
27-28	50,67	Indentifica versión y formato de DOS.
29-32	160	Constantes.
33-255		No usados.

Tabla 3: Formato de los Bloques de Directorio

BYTES	CONTENIDO
0-1	Track y sector del siguiente bloque de directorio.
2-31	Fichero número 1.
2	Tipo de Fichero: 0 - Fichero borrado. 1 - Fichero Secuencial (SEQ). 2 - Fichero Programa (PRG). 3 - Fichero Usuario (USR). 4 - Fichero Relativo (REL).
3-4	Track y sector del primer bloque del fichero.
5-20	Nombre del Fichero, rellenado con CHR$(160).
21-22	Track y sector del primer Side-Sector (Fichero REL).
23	Longitud del record (en Ficheros REL).
24-27	No usados.
28-29	Track y sector de para grabación con reemplazo.
30-31	Número de bloques usados por el fichero.
32-33	No usados.
34-65	FICHERO 2 (igual que bytes 2 a 33).
66-97	FICHERO 3 (igual que bytes 2 a 33).
98-129	FICHERO 4 (igual que bytes 2 a 33).
130-161	FICHERO 5 (igual que bytes 2 a 33).
162-193	FICHERO 6 (igual que bytes 2 a 33).
194-225	FICHERO 7 (igual que bytes 2 a 33).
226-255	FICHERO 8 (igual que bytes 2 a 31).



EL NUEVO COMMODORE 64 PORTATIL

Por P. MASATS

COMMODORE ha lanzado al mercado español una nueva versión de su conocido ORDENADOR PERSONAL C-64 que se caracteriza por incorporar un monofloppy 1541 de 170K y un monitor en color de 5 pulgadas además del ORDENADOR propiamente dicho. El nuevo modelo se llama SX-64.

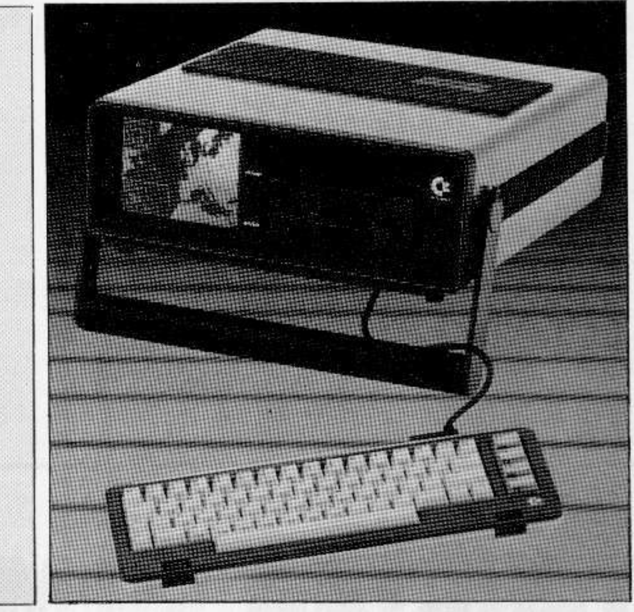
Este equipo es completamente compatible con el COMMODORE 64 tanto en lo que respecta al SOFTWARE como al HARDWARE. El diseño es muy cuidado para facilitar el transporte del ordenador allí donde se requiera la recogida de datos, realización de cálculos complejos, etc...

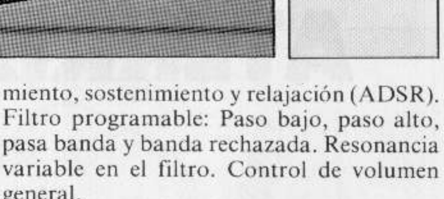
Como se puede ver en las fotografías adjuntas el aspecto exterior del ORDENADOR se ha cuidado al máximo para solventar el "lío de cables" que suele organizarse en la mesa de trabajo cuando se utiliza una configuración de CPU, UNIDAD DE DISCO y MONITOR, manteniendo un aspecto especialmente cuidado del conjunto. Merece destacarse la funcionalidad y ergonomía del teclado que durante el transporte se convierte en la tapa del equipo.

La calidad del monitor de video incorporado es excelente en especial cuando se trabaja en gráficos de alta resolución, no obstante, para un trabajo prolongado el equipo tiene salidas de video para monitores estandar en blanco y negro o color. El SX-64 lleva incorporados un amplificador y un altavoz para efectos de sonido y música generados por el sintetizador incorporado.

Las dos únicas diferencias entre el SX-64 y su antecesor el COMMODORE 64 son la ausencia en el primero de la conexión para la unidad de cassette (que queda sustituida con ventaja por la unidad de disco) y la ausencia en el SX-64 de una salida para televisor (en este caso es el monitor incorporado el que sustituye con ventaja a la opción del equipo original). La compatibilidad de los periféricos de ambos equipos es total, pudiendo utilizar el nuevo modelo la gama de periféricos disponibles para el COMMODORE 64.

En lo que se refiere al SOFTWARE el nuevo SX-64 ejecuta todos los programas disponibles para el COMMODORE 64 tanto ya desarrollados como de futura aparición dado que son internamente iguales.





Características generales

* 64K RAM
* SEGUNDO PROCESADOR Z80 EN OPCION
* GRAFICOS TRIDIMENSIONALES (SPRITES)
* SINTETIZADOR MUSICAL
* GRAFICOS DE ALTA RESOLUCION (320×200) * 16 COLORES
* CAPACIDAD PARA LOS INTERFACES RS232C — E IEEE-488.
* SALIDA PARA AMPLIACION DE HI-FI EXTERNO.
* JUEGO COMPLETO DE CARACTERES PET-CBM.
* CONEXION PARA CARTUCHOS.
* PORT DE USUARIO PROGRAMABLE.
* 2 PORT PARA JOYSTICK O LAPIZ OPTICO.
* PORT PARA DOS JUEGOS DE PADDLES. * CPU 6510.
* TOTAL COMPATIBILIDAD CON EL LENGUAJE ASSEMBLER 6502.
* INTERFACE PARA PERIFERICOS.
* DISCOS Y CARTUCHOS DE SOFTWARE COMPATIBLES CON LOS DEL COMMODORE 64.
* POSIBILIDAD DE TRABAJO CON UN MONITOR EXTERNO.

Datos técnicos

* Pantalla: Monitor de color integrado, de 40 columnas, 25 filas, 16 colores, 255 combinaciones de color pantalla/marco. Un total de 255 caracteres incluyendo 124 formas gráficas. Hasta 8 SPRITES, definida por el usuario, cada uno con su propia prioridad en la pantalla para la realización de efectos tridimensionales. Detección de colisión entre aprites o con cualquier otra forma de dentro de la pantalla. Gráficos de alta resolución de 320 × 200 puntos.

* Sonido: 3 voces de 9 octavas cada una, 4 formas de onda: diente de sierra, triangular, cuadrada de ancho variable y ruido. Generador programable de ataque, decaimiento, sostenimiento y relajación (ADSR). Filtro programable: Paso bajo, paso alto, pasa banda y banda rechazada. Resonancia variable en el filtro. Control de volumen general.

* Microprocesador: MOS 6510 (compatible en software con el MOS 6502).

* Formato de los datos numéricos: coma flotante, mantisa de 9 dígitos, exponente de 2 dígitos, de -39 a +38.

* Area de memoria del sistema operativo: 20K ROM - 3K RAM

* Area del usuario: 38K RAM o 54K RAM si no se usa el intérprete BASIC.

* Floppy integrado: 170K, uso de disco de 35 pistas, simple densidad, simple cara y de cinco pulgadas y cuarto. 17-21 sectores por pista, 256 bytes por sector, 683 bloques (664 libres).

* Entradas/Salidas: Port de usuario con posibilidad de conexión del estandar RS232C (requiere un conversor de nivel para ser plenamente operativo). Port serie para impresora. Port para cartuchos (juegos y expansiones ROM). Dos ports para joystick/paddles/lápiz óptico para control de juegos.

* Lenguaje: Intérprete BASIC residente.

—Variables: Reales, enteras, de cadena.

—Variables de matriz: reales, enteras, de cadena multidimensionales.

* Generalidades:

—Tamaño: 368 × 368 × 127 m/m.

—Peso: 11 Kg.

—Tensión de alimentación 220 V. de corriente alterna.

COMPARTIENDO EXPERIENCIAS ENTRE AMIGOS



Esta sección está dedicada a la colaboración de todos nuestros lectores y está dividida en dos partes:

1) Programación: Programas y similares.

2) Magia: Trucos, sugerencias, etcétera (pág. 50).

Habrá premios y alicientes para todos los participantes (ver pág. 6).

Enviarnos vuestra dirección para que podáis poneros en contacto unos con otros. ¡Anímaros, chicos (...y chicas)!

ATERRIZA COMO PUEDAS Y AROS

C-64	Pedro OBRADOR - Arta (MALLORCA)

Me llamo Pedro Obrador y tengo 16 años. Hacía tiempo ya que deseaba enviaros una colaboración, pero a falta de una os mando tres. Dos de ellas son originales mías, mientras que la tercera (zombis) es una adaptación para el CBM 64 de un programa aparecido en la revista inglesa "Personal Computer World" de junio del pasado año, cuyo programa era para la gama Pet CBM.

Aterriza como puedas

Es un juego que consiste en descender de una nave nodriza, esquivar un campo de asteroides y aterrizar sobre una de las plataformas, para luego volver a la nave nodriza. Cada vez que se terminen las plataformas se pasará a otra pantalla más difícil. Al principio del programa saldrá en la pantalla "espera un momento" (15 sgs.) durante los cuales el ordenador crea nuevos caracteres. La nave se maneja con la "z" (izquierda), "x" (derecha) y "m" (propulsor) que acelera cuando subes o para cuando bajas, gastando el fuel. La nave nodriza se abre pulsando cualquier tecla.

(5) Inicializa sonido.

(10) Dimensiona memoria.

(20) Instrucción de repetición de tecla.

(50) Subrutina de caracteres programables.

(60-70) Inicialización de variables.

(80) Subrutina de dibujo de pantalla y dibujo aleatorio de asteroides.

(90) Pone a Ø (cero) el número de caracteres en el buffer de teclado.

(100-245) Movimiento de la nave nodriza y movimiento aleatorio de los asteroides.

(250-345) Movimiento de la nave.

(350-510) Analiza si la nave choca con algo.

(550-600) Mensaje de fin de juego.

Aros

Es un programa de gráficos para el C-64. El programa utiliza el Bit-map. Empieza preguntando el número de aros y el desplazamiento horizontal y vertical. En la pantalla se dibujarán tantos aros como le indiques, desplazándose uno a uno la cantidad de pixels que tú le indiques. Cada aro que se vaya dibujando tendrá un tamaño más reducido que el anterior, dando así bonitos efectos cónicos si se utiliza el número máximo de aros.

(3-7) Pide las variables.

(9-10) Entra el modo Bit-map de alta resolución.

(20) Limpia el mapa de bit.

(30) Coloca el color en ciano y negro.

(40) Bucle de número de aros.

(50-120) Bucle para pintar la circunferencia.

(78-110) Calcula y dibuja el resultado en el pixel correspondiente de la pantalla.

(125) Indica que el programa ha terminado cambiando el color de la esquina superior izquierda.

(130) Entra en un bucle sin fin. Para salir del modo bit-map: Run/Stop y Restore.

Nota: La tercera colaboración no podemos publicarla de momento al ser una "adaptación" de otra revista. ■

ATERRIZA COMO PUEDAS

```
5 S=54273:POKE54296,15:POKE54277,16:
POKE54278,248:POKE54276,33
10 POKE53281,8:DIMI(40):DIMH(40)
20 POKE650,255
30 Q$="[HOM][23CRSRD]"
40 PRINT"[CLR]"LEFT$(Q$,11)TAB(10)"[RED]
ESPERA[SPC]UN[SPC]MOMENTO"
50 GOSUB3000:PRINTLEFT$(Q$,15)TAB(11
)"PULSA[SPC]UNA[SPC]TECLA"
55 GETB$:IFB$=""THEN55
60 A=1:B=18:D=3:E=0:F=20:G=0
70 J=800:K=1:F=F+3:L=0
80 GOSUB2000
90 POKE198,0
100 GETA$:M=0:IFA$=""THENM=1
110 P=1145+B:P1=P
150 IFB>35THENA=-1
160 IFB<1THENA=1
170 POKES,0:PRINT"[HOM][2CRSRD]"TAB(
B)"[4SPC]":B=B+A
180 PRINT"[HOM][2CRSRD]"TAB(B)"[ERROR]
[SHIFTC][SHIFTD][SHIFTE][SHIFTF]"
200 N=INT(RND(1)*F)+1
210 IFH(N)>1704THENI(N)=-1
220 IFH(N)<1304THENI(N)=1
230 POKEH(N),32:H(N)=H(N)+I(N):POKEH
(N),66
240 IFH(N)=P1THENGOTO360
245 IFM=1THEN100
250 GETM$
260 IFM$="Z"THENP=P-1
270 IFM$="X"THENP=P+1
275 IFPEEK(P)<>32THEN350
280 IFJ<10THEN300
290 IFM$="M"THENP=P-40:POKEP,32:J=J-
10:PRINT"[HOM]"TAB(22)J
295 POKES,RND(1)*255
300 IFPEEK(P)<>32THEN350
310 POKEP1,32:POKEP,64:POKE54272+P,2

320 P1=P:IFK=1THENP=P+40:GOTO150
330 P=P-40
340 IFP<1344THENFORX=1TO150:NEXT:GOT
O250
345 GOTO150
350 IFPEEK(P)=64THEN310
360 GOSUB2600
370 IFPEEK(P)=68ORPEEK(P)=69THENK=1:
POKEP1,32:IFL>7THENPOKEP,32:E=E+INT(
10*C)+J:GOTO70
390 IFPEEK(P)=68ORPEEK(P)=69THEN90
400 IFPEEK(P)=58ORPEEK(P)=59ORPEEK(P
)=60THEN460
410 IFPEEK(P)=65ORPEEK(P)=61THEN470
420 IFPEEK(P)=63THENE=E+150:POKEP,32
:L=L+1:GOTO500
430 K=1:G=G+1:GOSUB2500
433 FORX=100TO0STEP-1:POKES,X:POKEP1
,64:POKEP1,32:NEXT:IFG=3THEN550
435 IFL>7THENE=E+INT(10*C)+J:GOTO70
450 GOTO90
460 E=E+50:POKEP-2,32:POKEP-1,32:POK
EP,32:POKEP+1,32:POKEP+2,32:L=L+1:GO
TO500
470 E=E+100:POKEP-1,32:POKEP,32:POKE
P+1,32:L=L+1
500 GOSUB2500:K=2
510 GOTO150
550 PRINTLEFT$(Q$,13)TAB(12)"[ERROR]
FIN[SPC]DEL[SPC]JUEGO"
560 PRINTLEFT$(Q$,15)TAB(12)"[ERROR]
[CRSRD]OTRA[SPC]VEZ[SPC](S/N)"
565 FORY=1TO3:FORX=1TO255:POKES,X:NE
XTX,Y:POKES,0
570 POKE198,0
580 GETC$:IFC$<>"S"ANDC$<>"N"THEN580

590 IFC$="S"THENGOTO60
600 IFC$="N"THENPRINT"[CLR][ERROR]":
POKE53272,21:POKE53281,6:END
2000 POKES,0:PRINT"[CLR][CRSRD][ERROR]
[2SPC]PUNTOS"TAB(9)E;TAB(17)"FUEL[2SPC]
800"TAB(29)"NAVES[2SPC]";3-G
2010 FORI=1104TO1143:POKEI,45:POKEI+
54272,11:NEXT
2020 E$="[ERROR][3SPC][SHIFTA]=[9SPC]
[SHIFTA]=[15SPC][SHIFTA]=
2030 C$="[CRSRD][ERROR][3SPC]?[10SPC]
?[8SPC]?[7SPC]?[7SPC]"
2040 F$="[ERROR][13SPC]:;<
2047 FORX=1TO255:POKES,X:NEXT:FORX=2
55TO1STEP-1:POKES,X:NEXT:FORX=1TO255
:POKES,X:NEXT:POKES,0
```



```
2050 PRINTLEFT$(Q$,23)F$
2060 PRINTLEFT$(Q$,24)E$
2070 PRINTLEFT$(Q$,25)C$
2080 FORX=1984TO2023:POKEX,62:POKE54
272+X,11:NEXT
2090 FORO=1TOF
2100 I=RND(1):IFI>.5THENI(O)=1
2105 IFI<=.5THENI(O)=-1
2110 H(O)=INT(RND(1)*1703)+1
2120 IFH(O)<1304THEN2100
2130 POKEH(O),66
2140 NEXT
2145 FORI=1303TO1704
2150 POKE54272+I,2
2155 POKES,RND(1)*255
2160 NEXT
2170 POKES,0:RETURN
2500 PRINT"[HOM]"TAB(9)E;TAB(22)J;"[SPC]
";TAB(36)3-G
2520 RETURN
2600 FORX=1TO50:POKES,X:NEXT:FORY=XT
O1STEP-1:POKES,Y:NEXT:POKES,0:RETURN

3000 POKE56334,PEEK(56334)AND254:POK
E1,PEEK(1)AND251
3010 FORI=0TO71
3020 FORH=0TO7
3030 POKE12288+I*8+H,PEEK(53248+I*8+
H)
3040 NEXTH:NEXTI
3050 POKE1,PEEK(1)OR4:POKE56334,PEEK
(56334)OR1
3060 POKE53272,(PEEK(53272)AND240)+1
2
3070 FORI=58TO70
3080 FORH=0TO7:READQ
3120 POKE12288+(8*I)+H,Q
3140 NEXTH:NEXTI
3200 DATA255,255,255,127,63,31,7,3
3210 DATA255,255,255,255,255,255,255
,255
3220 DATA255,255,254,254,254,254,252
,252
3230 DATA252,240,224,224,192,128,128
,128
3240 DATA4,12,28,28,62,126,255,255
3250 DATA255,127,254,126,126,60,28,1
2
3260 DATA24,24,60,60,90,126,90,195
3270 DATA255,127,127,63,63,127,127,2
55
3280 DATA56,126,127,223,255,247,126,
60
3283 DATA129,65,33,23,29,127,91,255
3285 DATA15,124,192,192,128,128,0,0
3290 DATA240,62,3,3,1,1,0,0
3295 DATA129,130,132,232,184,254,218
,255
3300 RETURN
```

AROS

```
3 INPUT"NUMERO[SPC]DE[SPC]AROS[SPC](
1-16)";A
5 INPUT"DESPLAZAMIENTO[SPC]HORIZONTA
L";B
7 INPUT"DESPLAZAMIENTO[SPC]VERTICAL"
;D
9 BASE=2*4096:POKE53272,PEEK(53272)O
R8
10 POKE53265,PEEK(53265)OR32
20 FORI=BASETOBASE+7999:POKEI,0:NEXT

30 FORI=1024TO2023:POKEI,3:NEXT
40 FORC=0TOA-1
50 FORN=0TO2*πSTEPπ/180
60 X=INT(119+B*C+(80-5*C)*COS(N)):Y=
INT(100+D*C+(80-5*C)*SIN(N))
75 IFY<0ORY>199ORX<0ORX>319THEN120
78 CH=INT(X/8)
80 RO=INT(Y/8)
85 LN=YAND7
90 BY=BASE+RO*320+8*CH+LN
100 BI=7-(XAND7)
110 POKEBY,PEEK(BY)OR(2↑BI)
120 NEXTN,C
125 POKE1024,16
130 GOTO130
```



* **Ensamblador (assembler).** Lenguaje/programa que facilita la programación en código-lenguaje máquina.

* **Fichero (File).** Conjunto de datos, pueden ser almacenados en memoria RAM, en disco, cinta, o cualquier otro medio de almacenamiento.

* **Medio de almacenamiento (storage media).** Dispositivo físico en el que se almacenan datos o programas.

* **Página.** 256 posiciones de memoria consecutivas; la parte alta de sus direcciones es la misma en todas ellas.

* **Página cero.** La primera página de la memoria del ordenador, son las posiciones de memoria $ØØØØ a $ØØFF (Ø a 255 en decimal).

* **Procesador de textos.** Programa que permite utilizar el ordenador como una máquina de escribir electrónica con la ventaja adicional de poder corregir, trasladar, añadir, etc., párrafos antes de imprimirlos.

* **Salto (branch).** Alteración del contador de programa para que continúe su ejecución en otro lugar.

"RESOLUCION DE ECUACIONES"

VIC-20 + SUPERXPANDER CON 3K RAM	FELCA - CARTAGENA

Una línea

1Ø INPUT A,B,C,D,E,F:T=A*E-B*D:R=C*E-B*F:S=A*F-C*D:X=R/T:Y=S/T: PRINT; Y: GOTO 1Ø.

Este miniprogramita te pide en primer lugar los coeficientes de X,Y, término independiente de las dos ecuaciones $\left.\begin{matrix}Ax+By=C\\Dx+Ey=F\end{matrix}\right\}$ con dos incógnitas, luego calcula y por último te da los valores de "x" e "y", volviendo a pedir datos de nuevo.

Si tienes el VIC-20 con Super-expander hazlo así:

KEY 3,"1Ø INPUT A... (todo lo demás)... PRINTX;Y"+CHR$ (13)

Pulsando la F3 tendrás el programa a tu disposición sin pulsar RETURN.

Muchos juegos... pocos números

Os mando una cinta con muchos números. Repaso, resolución y gráfica (Superexpander VIC-20) de sistemas de ecuaciones de primer grado con dos incognitas.(El mismo que en UNA LINEA, pero con más.)

Decís que devolvéis la cinta con otro programa, bueno, pero por favor JUEGO no, si es posible que sea uno para crear mis propios caracteres, y podéis hacer vuestro éste y los sucesivos que prometo enviaré, claro, de NUMEROS, de esta forma podrán beneficiarse todos los suscriptores de COMMODORE WORLD; yo me siento "pagado" con el programa de regalo. GRACIAS.

Paso a comentar el programa:

Su utilización está orientada a la enseñanza (EGB y BUP), para lo cual se presenta un menú para utilizar opcionalmente el repaso (alumno) o la enseñanza (profesor) de los métodos de resolución: IGUALACION, SUSTITUCION, REDUCCION y REGLA DE CRAMER; y finalmente la solución GRAFICA y ANALITICA, de cualquier sistema, tan sólo con introducir los coeficientes, en orden, de x, y, término independiente de las dos ecuaciones (seis en total).

La solución gráfica se da en el entorno (-10,+10).

Las sentencias 10 a 30 son para presentar el programa, 90 a 100 es una subrutina camuflada, 110 a 170 es el menú, 180 a 190 es el estudio y paso a la opción marcada, 200 a 590 se muestran los repasos con ejemplos de los distintos métodos de resolución.

En la sentencia 1000 comienza el programa propiamente dicho y es para introducir los coeficientes de las dos ecuaciones, todos seguidos (separados por comas, claro está) los seis. En 1010 hace el estudio de compatibilidad de las ecuaciones y de no tener solución te lo indica (evitando mensajes de error). En 1020 calcula los determinantes (algoritmo empleado) y en 1050 a 1060 te calcula las soluciones, las redondea en las milésimas y las prepara para introducir en el modo GRAPHIC 2.

De 1100 a 1220 calcula el gráfico y te lo da en pantalla ayudado de la subrutina 5000 que calcula las rectas punto a punto.

Lo siento por los usuarios del C-64, pero sólo he llegado a poseer el VIC-2 y de los de antes, caros. ■

RESOLUCION DE ECUACIONES

```
5 REM"FELCA[COMM+]CARTAGENA"
10 PRINT"[CLR][CRSRD][5SPC]RESOLUCIO
N[SPC]DE":PRINT"[CRSRD]SISTEMAS[SPC]
DE[SPC]ECUACIONES"
20 PRINT"[2SPC]CON[SPC]DOS[SPC]INCOG
NITAS"
30 PRINT"[3CRSRD][SPC]AX+BY[SPC]=[SPC]
C[CRSRU][SHIFTI][CRSRL][CRSRD][SHIFT-]
[CRSRL][CRSRD][COMMQ]":PRINT"[SPC]DX
+EY[SPC]=[SPC]F[SHIFT-][CRSRL][CRSRD]
[SHIFTK][5CRSRD]"
90 PRINT"[3SPC]PARA[SPC]SEGUIR[SPC]P
ULSA":PRINT"[CRSRD][7SPC][COMM+]RETU
RN[COMM+]";
100 GETA$:IFA$<>CHR$(13)THEN100
110 PRINT"[CLR][CRSRD][5SPC][COMM+]O
PCIONES[COMM+]"
120 PRINT"[CRSRD][COMM-]1-REPASAR[SPC]
METODO[SPC]DE":PRINT"[5SPC][COMM£]IG
UALACION[COMM£]"
130 PRINT"[CRSRD][COMM-]2-REPASAR[SPC]
METODO[SPC]DE":PRINT"[5SPC][COMM£]SU
STITUCION[COMM£]"
140 PRINT"[CRSRD][COMM-]3-REPASAR[SPC]
METODO[SPC]DE":PRINT"[5SPC][COMM£]RE
DUCCION[COMM£]"
150 PRINT"[CRSRD][COMM-]4-REPASAR[SPC]
METODO[SPC]DE":PRINT"[3SPC][COMM£]RE
GLA[SPC]DE[SPC]CRAMER[COMM£]"
160 PRINT"[CRSRD][COMM-]5-RESOLUCION
[SPC]GRAFICA":PRINT"[5SPC]Y[SPC]ANAL
ITICA"
170 PRINT"[3CRSRD][2SPC]MARCA[SPC]TU
[SPC]OPCION":INPUT"AQUI";O
180 IFO<1ORO>5THEN110
190 ONOGOTO200,300,400,500,1000
200 PRINT"[CLR][COMM+]METODO[SPC]DE[SPC]
IGUALACION[COMM+]"
210 PRINT"TENEMOS[SPC]QUE[SPC]DESPEJ
AR":PRINT"UNA[SPC]DE[SPC]LAS[SPC]INC
OGNITAS"
220 PRINT"[SPC]LUEGO[SPC]IGUALAMOS[SPC]
LOS":PRINT"[2SPC]VALORES[SPC]OBTENID
OS"
230 PRINT"[4SPC][2COMM+]EJEMPLO[2COMM+]
"
240 PRINT"[CRSRD][SPC]X+5Y[SPC]=[SPC]
13[CRSRU][SHIFTI][CRSRD][CRSRL][SHIFT-]
[SPC]X[SPC]=[SPC]13-5Y"
250 PRINT"[CRSRD][SPC]X-[SPC]Y[SPC]=
[2SPC]7[CRSRU][COMMQ][CRSRD][CRSRL][SHIFT-]
[CRSRL][CRSRD][SHIFTK][CRSRU][SPC]X[SPC]
=[2SPC]7+[SPC]Y"
260 PRINT"[CRSRD][SPC]13-5Y[SPC]=[SPC]
7+Y[SPC],[SPC]LUEGO"
270 PRINT"[CRSRD][2SPC]6Y[SPC]=[SPC]
6[2SPC]=>[2SPC][COMM+]Y=1[COMM+]"
280 PRINT"[CRSRD]SUSTITUYENDO[SPC]Y[SPC]
POR[SPC]1":PRINT"TENEMOS[SPC]EL[SPC]
VALOR[COMM+]X=8[COMM+][CRSRD]"
290 GOTO90
300 PRINT"[CLR]METODO[SPC]DE[SPC]SUS
TITUCION"
310 PRINT"[CRSRD]TENEMOS[SPC]QUE[SPC]
DESPEJAR":PRINT"UNA[SPC]DE[SPC]LAS[SPC]
INCOGNITAS"
320 PRINT"[2SPC]LUEGO[SPC]SUSTITUIMO
S":PRINT"[SPC]EL[SPC]VALOR[SPC]EN[SPC]
LA[SPC]OTRA[SPC]"
330 PRINT"ECUACION[2SPC][2COMM+]EJEM
PLO[2COMM+]"
340 PRINT"[CRSRD][2SPC]X+Y[SPC]=[SPC]
19[CRSRU][SHIFTI][CRSRD][CRSRL][SHIFT-]
[SPC]Y[SPC]=[SPC]19-X"
350 PRINT"[CRSRD][SPC]2X-Y[SPC]=[2SPC]
2[CRSRU][COMMQ][CRSRD][CRSRL][SHIFT-]
[CRSRD][CRSRL][SHIFTK][CRSRU]2X-(19-
X)=2"
```

▶

```
360 PRINT"[CRSRD]DESPEJANDO[COMM+]X[COMM+]
,TENEMOS":PRINT"[SPC]3X[SPC]=[SPC]21
[2SPC]=>[2SPC][COMM+]X=7[COMM+]"
370 PRINT"[CRSRD]SUSTIYUYENDO[SPC]X[SPC]
POR[SPC]7":PRINT"TENEMOS[SPC]EL[SPC]
VALOR[COMM+]Y=12[COMM+]"
390 GOTO90
400 PRINT"[CLR][COMM+]METODO[SPC]DE[SPC]
REDUCCION[COMM+]"
410 PRINT"[CRSRD]TENEMOS[SPC]QUE[SPC]
CONSEGUIR":PRINT"QUE[SPC]LOS[SPC]COE
FICIENTES":PRINT"DE[SPC]UNA[SPC]MISM
A[SPC]VARIABLE"
420 PRINT"SEAN[SPC]IGUALES[SPC]Y[SPC]
SIGNO":PRINT"CONTRARIO;AHORA[SPC]SUM
AR"
430 PRINT"[4SPC][2COMM+]EJEMPLO[2COMM+]
"
440 PRINT"[CRSRD][SPC]X+5Y[SPC]=[SPC]
13[CRSRU][SHIFTI][CRSRD][CRSRL][SHIFT-]
"
450 PRINT"[CRSRD][SPC]X-[SPC]Y[SPC]=
[2SPC]7[CRSRU][COMMQ][CRSRD][CRSRL][SHIFT-]
[CRSRL][CRSRD][SHIFTK][CRSRU][SPC][SHIFTV]
[SPC]5,NOS[SPC]DA"
460 PRINT"[CRSRD][SPC]X+5Y[SPC]=[SPC]
13[CRSRU][SHIFTI][CRSRD][CRSRL][SHIFT-]
"
470 PRINT"[CRSRD]5X-5Y[SPC]=[SPC]35[CRSRU]
[COMMQ][CRSRD][CRSRL][SHIFT-][CRSRL]
[CRSRD][SHIFTK][2CRSRU]+=[SPC]6X[SPC]
=[SPC]48[CRSRD][7CRSRL][COMM+]X[SPC]
=[SPC]8[COMM+]"
480 PRINT"[CRSRD]SUSTITUYENDO[SPC]X[SPC]
POR[SPC]8":PRINT"TENEMOS[SPC]EL[SPC]
VALOR[COMM+]Y=1[COMM+]"
490 GOTO90
500 PRINT"[CLR][2SPC][COMM+]REGLA[SPC]
DE[SPC]CRAMER[COMM+]"
510 PRINT"[CRSRD][SPC]AX+BY[SPC]=[SPC]
C[CRSRU][SHIFTI][CRSRL][CRSRD][SHIFT-]
[CRSRL][CRSRD][COMMQ]":PRINT"[SPC]DX
+EY[SPC]=[SPC]F[SHIFT-][CRSRL][CRSRD]
[SHIFTK]"
520 PRINT"[CRSRD]FORMAMOS[SPC]DETERM
INANTES":PRINT"[3SPC][SHIFT-]AB[SHIFT-]
[3SPC][SHIFT-]AC[SHIFT-][3SPC][SHIFT-]
CB[SHIFT-]"
530 PRINT"[SPC]T=[SHIFT-][2SPC][SHIFT-]
[SPC]S=[SHIFT-][2SPC][SHIFT-][SPC]R=
[SHIFT-][2SPC][SHIFT-]":PRINT"[3SPC]
[SHIFT-]DE[SHIFT-][3SPC][SHIFT-]DF[SHIFT-]
[3SPC][SHIFT-]FE[SHIFT-]"
540 PRINT"[CRSRD][SPC]VALORES[SPC]DE
[SPC]VARIABLES"
550 PRINT"[2CRSRD][2SPC][COMM+]X[COMM+]
=[SPC][SHIFT*][CRSRU][CRSRL]R[2CRSRD]
[CRSRL]T[CRSRU][5SPC][COMM+]Y[COMM+]
=[SPC][SHIFT*][CRSRU][CRSRL]S[2CRSRD]
[CRSRL]T"
590 GOTO90
1000 PRINT"[CLR]MARCA[SPC]LOS[SPC]CO
EFICIENTES":INPUTA,B,C,D,E,F
1010 IFA/D=B/ETHENPRINT"[SPC]SON[SPC]
RECTAS[SPC]PARALELAS[4CRSRD]":GOTO90

1020 T=A*E-B*D:R=C*E-B*F:S=A*F-C*D
1050 X=R/T:X=INT(X*1000+.5)/1000:X$=
"X="+STR$(X)
1060 Y=S/T:Y=INT(Y*1000+.5)/1000:Y$=
"Y="+STR$(Y)
1070 GRAPHIC2:DRAW2,512,0TO512,1023:
DRAW2,0,512TO1023,512
1100 FORK=6TO975STEP51:POINT2,K,524:
NEXTK:FORJ=6TO975STEP51:POINT2,524,J
:NEXTJ
1140 GOSUB5000
1150 A=D:B=E:C=F
1160 GOSUB5000
1170 CHAR0,9,"Y":CHAR9,19,"X":CHAR0,
0,X$:CHAR1,0,Y$
1210 GETA$:IF A$<>CHR$(13)THEN1210
1220 GRAPHIC0:GOTO110
5000 FORV=10TO-10STEP-.125:U=((C-B*V
)/A+10)*51:IFU<0ORU>1023THEN5050
5030 Z=(10-V)*51:POINT2,U,Z
5050 NEXTV:RETURN
```

BOGGLE

C-64

Juan Carlos LOREN - Leganés (MADRID)

Os envío este programa al que he llamado "Boggle", por su parecido con dicho juego. El programa se habría podido ampliar a dos jugadores o haberlo variado de muchas formas y así hacerlo más completo, pero he creído más conveniente dejarlo así, porque ampliarlo demasiado puede dar lugar a que más de uno de se asuste, al ver demasiado "rollo" impreso para un simple juego, teniendo en cuenta lo pesado que es digitar un programa.



El manejo es de lo más sencillo

Al ejecutar RUN, en la parte izquierda de la pantalla se dibuja una matriz con unos puntos, que seguidamente se rellena con letras al azar gracias a la instrucción "RND". Una vez rellena dicha matriz podemos empezar el juego digitando alguna palabra. Las letras de dicha palabra serán impresas en la pantalla, sólo si se encuentran dentro de la matriz citada, a la vez que desaparecerán de ésta dibujando un espacio en su lugar. Si digitamos alguna letra que no exista se nos advertirá de este hecho, con el mensaje "letra inexistente" que aparecerá en la pantalla algunos segundos; igualmente si digitamos alguna tecla que no pertenezca al abecedario, aparecerá el mensaje "digite sólo letras". Cuando hayamos concluido la palabra, pulsaremos "return" y se nos dará nuestra puntuación hasta ese momento, a la vez que se nos preguntará si deseamos continuar. En caso de que le respondamos "S", podremos continuar digitando otras palabras con las letras que aún existan en la matriz.

Hay que advertir que las palabras no deben ser nunca mayores de dieciocho letras, porque en ese caso el programa salta a la rutina de fin igual que si hubiésemos pulsado "return". ■

BOGGLE

```
1 REM **********
2 REM * BOGGLE *
3 REM **********
5 DIMP(100):DIMQ(100)
10 PRINT"[CLR]"
20 A=1024
30 FORT=1TO20:POKEA,46:A=A+1:NEXTT
40 A=A+60:IFA>1823THENGOTO60
50 GOTO30
60 A=1065
70 FORT=1TO10:POKEA,46:A=A+2:NEXTT
80 A=A+60:IFA>1784THENGOTO100
90 GOTO70
100 A=1064:X=22:E=0
110 FORT=1TO10:P(E)=A
120 G=INT(1+RND(5)*26)
130 POKEA,G:A=A+2:Q(E)=G
140 E=E+1:NEXTT
150 A=A+60:IFA>1724THENGOTO170
160 GOTO110
170 PRINTTAB(22)"[RVSON]DIGITE[SPC]P
ALABRA[RVSOFF][CRSRD]"
190 A$=""
200 GETA$:IFA$=""THEN200
220 D=ASC(A$):E=0:IFD=13THENGOTO1500

240 IFD>90ORD<65THENGOTO1200
260 IFQ(E)=D-64THENGOTO300
280 E=E+1:IFE>90THENGOTO1300
290 GOTO260
300 PRINTTAB(X):X=X+1:PRINTA$:GOSUB1
100
320 W=P(E):POKEW,32:Q(E)=0:S=S+1:L=L
+1
340 IFS=18THENGOTO1500
360 GOTO190
1000 POKE781,22:POKE782,1:POKE783,0:
SYS65520 :RETURN
1100 POKE781,3:POKE782,1:POKE783,0:S
YS65520:RETURN
1200 GOSUB1000
1210 PRINT"DIGITE[SPC]SOLO[SPC]LETRA
S":FORT=1TO1000:NEXT
1220 FORT=1904TO1983:POKET,32:NEXT
1240 GOSUB1100
1250 GOTO190
1300 GOSUB1000
1310 PRINT"LETRA[SPC]INEXISTENTE":FO
RT=1TO1000:NEXT
1330 FORT=1904TO1983:POKET,32:NEXT
1340 GOSUB1100
1350 GOTO190
1500 GOSUB1000
1510 PRINT"SU[SPC]PUNTUACION[SPC]ES[SPC]
DE[SPC]"L"[SPC]PUNTOS":S=0
1520 PRINTTAB(1)"DESEA[SPC]CONTINUAR
[SPC](S/N)?"
1522 GETB$:IFB$=""THENGOTO1522
1525 IFB$="N"THENPRINT"[CLR]":END
1527 IFB$="S"THENGOTO1530
1528 IFB$<>"S"ORB$<>"N"THENB$="":GOT
O1522
1530 FORT=1904TO1979:POKET,32:NEXTT
1560 GOSUB1100
1570 FORT=1165TO1183:POKET,32:NEXTT
1580 X=22:GOTO190
```

UFO

VIC-20 — **René RODRIGUEZ LLEONART - Hospitalet (BARCELONA)**

1. El programa que os envío con la carta es lo último que he hecho y, aunque no destaca por su originalidad, sí tiene la ventaja o el aliciente de permitir que jueguen dos personas a la vez.

La puntuación depende de la distancia a la que se encuentre la vave y el número de misiles aumenta en uno por cada 75 puntos. El juego termina cuando los dos jugadores han disparado todos sus misiles y gana aquél que ha conseguido más puntos. Las bases de lanzamiento del jugador de la derecha corresponden a los números 7, 8 y 9 del teclado y los números 1, 2 y 3 son para el jugador de la izquierda.

El juego no tiene más secretos y pienso que por su sencillez podréis cambiar lo que queráis para amoldarlo a vuestro gusto. Tento 18 años. ■

P.D.: Se puede dar un poco más de rapidez al juego añadiendo la línea: 203 IFD<>1ANDL<>-1THEN120

UFO

```
1 POKE56,28:POKE52,28
2 FORT=7551TO7631:POKET,PEEK(T+25600
):NEXT
3 FORT=7424TO7431:POKET,PEEK(T+25600
):NEXT
4 GOSUB1000
5 POKE36878,15:V=7680:Q=4:U=6:K1=4:K
2=6:DY=0
6 PRINT"[CLR]"
10 POKE36879,238
20 A(1)=7813:A(2)=7923:A(3)=8033
30 B(7)=7854:B(8)=7964:B(9)=8074
40 FORT=1TO9:POKEA(T),3:POKEB(T),5:P
OKEA(T)+30720,2:POKEB(T)+30720,2:NEX
TT
100 Z=INT(RND(0)*2)+1:DY=1:IFZ=1THEN
P=484:DY=-1
110 X=INT(RND(0)*10)+6
120 POKEV+X+P+Y*22,1:POKEA(Q)+D,0:PO
KEB(U)+L,2:POKE36874,240:GOSUB270:PO
KEV+X+P+Y*22,32
121 POKEB(U)+L,32:POKEA(Q)+D,32:POKE
36874,0
130 IFD>18THEND=0:Q=0:IFK1=4THENIFE1
=0THENGOSUB400
140 IFL<-18THENL=0:U=6:IFK2=6THENIFE
2=0THENGOSUB400
```

```
150 GETA$:IFD=0THENIFVAL(A$)<K1THENQ
=VAL(A$):IFQ<>0THENC1=C1+1
185 IFL=0THENIFVAL(A$)>K2THENU=VAL(A
$):C2=C2+1
186 Y=Y+DY:IFQ<>0THEND=D+1
187 IFU<>6THENL=L-1
190 IFZ<>1THENIFY>23ORY<-3THENY=0:P=
0:GOTO100
191 IFZ=1THENIFY>0ORY<-22THENY=0:P=0
:GOTO100
200 E1=INT(PA/75)+10-C1:E2=INT(PB/75
)+10-C2
202 PRINT"[HOM][BLK][5CRSRR][2CRSRD]
";PA"[2CRSRR][2SPC]"PB
205 PRINT"[17CRSRD][BLK][2CRSRR][2SPC]
[3CRSRL]"E1:PRINT"[3CRSRL][2SPC][3CRSRL]
"E2
210 GOTO120
270 IFPEEK(V+X+Y*22+P)=0THENI=0:GOTO
500
280 IFPEEK(V+X+Y*22+P)=2THENI=3:GOTO
500
281 RETURN
400 POKE36869,240:PRINT"[CRSRU][5CRSRR]
[YEL]NO[SPC]MISSILS!!!"
405 FORP=1TO5:FORM=180TO235STEP3:POK
E36876,M:FORN=1TO5:NEXTN:NEXTM:POKE3
6876,0
410 FORM=1TO5:NEXTM:NEXTP
415 IFE1=0THENK1=0
```

```
420 IFE2=0THENK2=9
430 PRINT"[CRSRU][19SPC]":POKE36869,
255
440 IFE1=0ANDE2=0THENPOKE36869,240:E
ND
450 RETURN
499 POKE36878,15
500 POKEV+X+22*Y+P,4:POKE36877,220:F
ORN=15TO0STEP-1:POKE36878,N:FORM=1TO
300:NEXTM
501 NEXTN:POKE36877,0:POKE36878,15
509 POKEV+X+Y*22+P,2
510 IFI=0THENPA=PA+10*D:Q=4:D=0
515 IFI=3THENPB=PB-10*L:U=6:L=0
520 POKEV+X+22*Y+P,32:Y=0
525 RETURN
1000 FORI=7168TO7215
1001 READA:POKEI,A
1002 NEXTI
1010 DATA160,56,190,63,190,56,160,0
1015 DATA24,60,231,126,24,60,102,195

1024 DATA 0,5,28,125,252,125,28,5
1025 DATA252,252,12,30,30,12,252,252

1026 DATA24,24,24,255,255,24,24,24
1027 DATA126,126,96,240,240,96,126,1
26
1040 POKE36869,255
1050 RETURN
```

TRAGAPERRAS

C-64	Por Antonio Font

Como su nombre dice es como las máquinas de los bares.

El juego consiste: Primero, al teclear RUN, en pantalla sale ECHE MONEDA y "pulsar P", luego se borra la pantalla e indica ,en dos recuadros "STOP" y "START" y un signo parpadeando al lado de Start. Pulsamos S y el signo pasa al lado de Stop y empieza a dar vueltas el primer tambor.

Ahora hay dos opciones, que lo paremos nosotros o que pare solo.

Si lo paramos nosotros, pulsando S, o se para sola pasa al siguiente tambor hasta llegar al tercero que al pararlo pueden haber 2 opciones.

1.—Que salgan avances, y que se pueda avanzar un tambor pulsando el número de su columna, o si no se quiere avanzar, pulsar 4. Una vez hecho esto el VIC-20 da los resultados.

2.—Que dé los resultados directamente. Según los resultados, de la página siguiente, puede dar dos opciones.

1.—Que haya tocado uno de los resultados y entonces indica en la pantalla el dinero que ha tocado, el dinero que tiene en total y si quiere continuar o no.

2.—Si no le ha tocado ningún resultado entonces puede:

2a. si tiene dinero en el bote indica el dinero total y si quiere continuar.

2b. Si no tiene dinero indica que en el bote hay 0 pts. y GAME OVER.

El programa es un pcoo complicado de explicar pero si se prueba, en seguida se entiende.

Si tuviera ampliación hubiera hecho que hasta que la máquina no estuviera llena no diese premios, hubiera puesto música y más diversiones.

TRAGAPERRAS

```
1 REM ANTONIO FONT
2 REM  BARCELONA
3 REM
8 POKE36879,28
10 PRINT"[CLR]"
11 C=1:Z=1
12 PRINT"[HOM][10CRSRD][3CRSRR][RED]
HECHE[SPC]25[SPC]PTS":PRINT"[3CRSRR]
[2CRSRD][GRN]PULSE[RVSON][RED]P[RVSOFF]
[BLU]"
13 GETA$:IFA$="P"THEN15
14 GOTO13
15 POKE36878,15:FORX=135TO241:POKE36
877,X:NEXT:A=25:POKE36877,0
16 PRINT"[CLR]"
20 PRINT"[HOM][CRSRD][SPC][YEL][RVSON]
[6SPC][RVSOFF][3SPC][RED][RVSON][7SPC]
[RVSOFF]"
21 PRINT"[SPC][YEL][RVSON][SPC][RVSOFF]
[BLK]STOP[YEL][RVSON][SPC][RVSOFF][3SPC]
[RED][RVSON][SPC][RVSOFF][BLK]START[RED]
[RVSON][SPC][RVSOFF][BLU]"
22 PRINT"[SPC][YEL][RVSON][6SPC][RVSOFF]
```

```
[3SPC][RED][RVSON][7SPC][RVSOFF][BLU]
"
30 GETA$:IFA$="S"THEN50
31 PRINT"[HOM][2CRSRD][18CRSRR][RED]
[SHIFTQ]";:POKE36876,200:FORX=1TO100
:NEXTX
32 POKE36876,0
33 PRINT"[CRSRL][SPC]"
40 GOTO30
50 C=C+1:IFC=30THENZ=8:M=P
51 U=INT((RND(1)*10)+1):F=F+1:IFF<>1
00THENGOTO60
52 IFF=100THENPRINT"[HOM][15CRSRD]"
53 IFZ=1THENM=P
54 IFZ=8THENN=P
55 IFZ=15THENO=P
56 GOTO590
60 PRINT"[HOM][2CRSRD][8CRSRR][YEL][SHIFTQ]
[BLU]";:FORX=187TO200:POKE36876,X:NE
XTX
61 PRINT"[CRSRL][SPC]":POKE36876,0:I
FC>30THEN63
62 GETB$:IFB$="S"ANDC<=30THENZ=8:C=3
0:M=P
63 IFC>60THEN65
64 GETB$:IFB$="S"ANDC>30THENZ=15:C=6
0:N=P
65 IFC=60THENZ=15:N=P
66 GETB$:IFB$="S"THENO=P:GOTO562
67 IFC=90THENO=P:GOTO562
68 PRINT"[HOM][5CRSRD]";
69 ONUGOTO70,120,170,220,270,320,370
,420,470,520
70 PRINTSPC(Z)"[SHIFTO][4COMMT][SHIFTP]
"
80 PRINTSPC(Z)"[COMMG][SPC][RVSON][CYN]
[SHIFT£][COMM*][RVSOFF][BLU][SPC][COMMM]
"
90 PRINTSPC(Z)"[COMMG][SPC][CYN][COMM*]
[SHIFT£][BLU][SPC][COMMM]"
100 PRINTSPC(Z)"[SHIFTL][4COMM@][SHIFT@]
"
110 P=1:GOTO50
120 PRINTSPC(Z)"[SHIFTO][4COMMT][SHIFTP]
"
130 PRINTSPC(Z)"[COMMG][SPC][BLK][RVSON]
[2COMM+][RVSOFF][BLU][SPC][COMMM]"
140 PRINTSPC(Z)"[COMMG][SPC][BLK][RVSON]
[2COMM+][RVSOFF][BLU][SPC][COMMM]"
150 PRINTSPC(Z)"[SHIFTL][4COMM@][SHIFT@]
"
160 P=2:GOTO50
170 PRINTSPC(Z)"[SHIFTO][4COMMT][SHIFTP]
"
180 PRINTSPC(Z)"[COMMG][SPC][PUR][RVSON]
[SHIFTU][SHIFTI][RVSOFF][BLU][SPC][COMMM]
"
190 PRINTSPC(Z)"[COMMG][SPC][PUR][RVSON]
[SHIFTJ][SHIFTK][RVSOFF][BLU][SPC][COMMM]
"
200 PRINTSPC(Z)"[SHIFTL][4COMM@][SHIFT@]
"
210 P=3:GOTO50
220 PRINTSPC(Z)"[SHIFTO][4COMMT][SHIFTP]
"
230 PRINTSPC(Z)"[COMMG][RED][RVSON][SHIFTO]
[2COMMT][SHIFTP][RVSOFF][BLU][COMMM]
"
240 PRINTSPC(Z)"[COMMG][RED][RVSON][SHIFTL]
[2COMM@][SHIFT@][RVSOFF][BLU][COMMM]
"
250 PRINTSPC(Z)"[SHIFTL][4COMM@][SHIFT@]
"
260 P=4:GOTO50
270 PRINTSPC(Z)"[SHIFTO][4COMMT][SHIFTP]
```

```
"
280 PRINTSPC(Z)"[COMMG][SPC][RVSON][GRN]
[SHIFT£][COMM*][RVSOFF][BLU][SPC][COMMM]
"
290 PRINTSPC(Z)"[COMMG][SPC][GRN][SHIFT£]
[COMM*][BLU][SPC][COMMM]"
300 PRINTSPC(Z)"[SHIFTL][4COMM@][SHIFT@]
"
310 P=5:GOTO50
320 PRINTSPC(Z)"[SHIFTO][4COMMT][SHIFTP]
"
330 PRINTSPC(Z)"[COMMG][BLK]$[2SPC][YEL]
$[BLU][COMMM]"
340 PRINTSPC(Z)"[COMMG][CYN]$[2SPC][RED]
$[BLU][COMMM]"
350 PRINTSPC(Z)"[SHIFTL][4COMM@][SHIFT@]
"
360 P=6:GOTO50
370 PRINTSPC(Z)"[SHIFTO][4COMMT][SHIFTP]
"
380 PRINTSPC(Z)"[COMMG][RED][SHIFTS]
[SPC][SPC][SHIFTS][BLU][COMMM]"
390 PRINTSPC(Z)"[COMMG][RED][SHIFTS]
[2SPC][SHIFTS][BLU][COMMM]"
400 PRINTSPC(Z)"[SHIFTL][4COMM@][SHIFT@]
"
410 P=7:GOTO50
420 PRINTSPC(Z)"[SHIFTO][4COMMT][SHIFTP]
588 IFV$="3"THENZ=15:C=60:F=98:GOTO5
0
```

(Pasa a pág. 44)

Cuando se es COMMODORE es muy difícil ser modesto

COMMODORE 64

Cuando se tiene 64 K de memoria, una magnífica resolución, 16 colores, efectos tridimensionales con sprites, un sonido equivalente al de un sintetizador, un teclado profesional con 62 caracteres gráficos, toda una amplia gama de periféricos, la más completa gama de programas educativos, profesionales y de video-juegos...; en resumen, cuando se es un ordenador personal como no existe ningún otro en el mercado y el más vendido mundialmente, es muy difícil decir sin orgullo que eres un Commodore-64.

Claro que más difícil todavía es decir sin orgullo que tienes un Commodore-64. ¿Por qué no lo comprueba?

SLOGAN

COMMODORE 64 LE DA ACCESO A MUCHOS ACCESORIOS

Unidad simple de disco (Monofloppy) 170 K.
Cassette.



Plotter e impresora, 4 colores, 14 c.p.s.

Impresora matricial, tractor, 30 c.p.s.

Interface RS232.

Joy Stick.

Paddle.

Cursos de Introducción al BASIC.

COMMODORE 64 LE MUESTRA PARTE DE SUS PROGRAMAS

Utilitarios y lenguajes

MONITOR LENGUAJE MAQUINA.
FORTH.
LOGO.
PILOT.
MACRO ASSEMBLER.
PROGRAMMER'S UTILITIES.
TURTLE GRAPHICS II.
MASTER.

Sistemas operativos

FILE/BOSS.
CP/M.

Programas de aplicaciones

EASY SCRIPT.
Proceso de texto de gran potencia.

CALC RESULT.
Hoja electrónica de cálculo.

EASY CALC RESULT.
Versión simplificada del CALC RESULT.

MAGIC DESK.
Proceso de texto y gestión de ficheros.

AGENDA TELEFONICA.

Programas educativos

MUSIC MACHINE.
MUSIC COMPOSER.
VISIBLE SOLAR SYSTEM.
SPEED/BINGO MATH.
FISICA I.
MATEMATICAS I.
HISTORIA I.
GEOGRAFIA I.
GEOGRAFIA II.
JUEGOS EDUCATIVOS.
TEMAS MONOGRAFICOS.
CONOCIMIENTOS GENERALES.
QUIMICA I.

Juegos

JUPITER LANDER.
KICKMAN.
SEAWOLF.
RADAR RAT RACE.
TOOTH INVADERS.
LAZARIAN.
OMEGA RACE.
LE MANS.
PINBALL SPECTACULAR.
AVENGER.
SUPERMASH.
FROGMASTER.
GRID RUNNER.
ATTACK OF THE MUTANT CAMELS.
THE PIT.
MR. TNT.
6 GAME PROGRAMS.
BINGO.
ROOTING TOOTING.
MINESSOTA FAT'S POOL CHALLENGE.

... y seguimos ampliando la lista

El ordenador personal de la familia más potente



MICROELECTRONICA Y CONTROL, S.A.
c/. Taquígrafo Serra, 7, 5.°. Barcelona-29
c./ Princesa, 47, 3.° G. Madrid-8

(Viene de pág. 41.)

```
590 IFI=5THENPRINT"[5CRSRU][14CRSRR]
4[3CRSRR][RVSON][YEL][SPC][RVSOFF]":
I=4:GOTO583
591 IFI=4THENPRINT"[5CRSRU][10CRSRR]
3[3CRSRR][RVSON][YEL][SPC][RVSOFF]":
I=3:GOTO583
592 IFI=3THENPRINT"[5CRSRU][6CRSRR]2
[3CRSRR][RVSON][YEL][SPC][RVSOFF]":I
=2:GOTO583
593 IFI=2THENPRINT"[5CRSRU][2CRSRR]1
[3CRSRR][RVSON][YEL][SPC][RVSOFF]":I
=1:GOTO583
594 IFI=1THENPRINT"[5CRSRU][2CRSRR][RVSON]
[YEL][SPC][RVSOFF]":I=0:GOTO583
600 IFM=6ANDO=MANDN=MTHENW=W+2500:V=
100:GOTO800
610 IFM=10ANDN=MANDO=MTHENW=W+450:V=
18:GOTO800
620 IFM=7ANDN=MANDO=MTHENW=W+300:V=1
2:GOTO800
630 IFM=3ANDN=MANDO=6THENW=W+250:V=1
0:GOTO800
631 IFM=5ANDM=NORO=5ANDO=NTHENW=W+50
:V=2:GOTO800
633 IFM=6ANDO=7ANDN=MTHENW=W+1000:V=
40:GOTO800
634 IFN=7ANDM=NTHENW=W+500:V=20:GOTO
800
640 IFM=NANDN=OTHENW=W+100:V=4:GOTO8
00
645 IFM=NORN=OTHENW=W+75:V=3:GOTO800
```

NOTICIAS COMMODORIANAS

Los Commodore por los Ministerios

El Centro de Proceso de Datos del Ministerio de Educación y Ciencia, ha conectado a modo de ensayo un VIC-20 y un C-64 en modo TTY (terminales teletipo) al UNIVAC DTC 1100/80 el ordenador central del Ministerio para toda España. La prueba fue todo un exitazo y no digamos nada del orgullo que sentimos viendo a nuestro inefable VIC y nuestro no menos idolatrado 64 siguiendo tan marchosamente el ritmo del gran gigante (y sin perder el paso por un segundo).

¿Qué otras sorpresas nos tienen guardadas en cartera nuestros dos queridos "aparatitos"...?

```
670 IFM=5ORO=5THENW=W+25:V=1:GOTO800
700 GOTO900
800 PRINT"[HOM][15CRSRD][22SPC][21CRSRL]
";
810 FORJ=0TOV:FORL=128TO255STEP11:PO
KE36874,L
820 B=B+1:IFB=3THENB=0:POKE36874,0
830 NEXTL:PRINT"[RED][HOM][15CRSRD][2CRSRR]
"H"[SPC]""PTS":H=H+25:NEXTJ:H=0:FORG
=1TO500:NEXTG
840 GOTO910
900 PRINT"[HOM][15CRSRD][22SPC][21CRSRL]
[2CRSRD]";
910 PRINT"[RED]DINERO[SPC]QUE[SPC]TI
ENE[SPC]EN[3SPC][CRSRD]EL[SPC]BOTE"W
"PTS"
920 IFW=0THENPRINT"[CRSRD][GRN][SPC]
GAME[SPC]OVER":FORG=1TO1000:NEXTG:GO
TO8
930 PRINT"[CRSRD][GRN]QUIERE[SPC]CON
TINUAR?"
940 GETE$:IFE$="S"THENW=W-25:GOTO8
950 IFE$="N"THENPOKE36879,27:PRINT"[BLU]
[CLR]":END
960 GOTO940
"
430 PRINTSPC(Z)"[COMMG][RED][COMMD][COMMF]
[YEL][COMMD][COMMF][BLU][COMMM]"
440 PRINTSPC(Z)"[COMMG][YEL][COMMC][COMMV]
[RED][COMMC][COMMV][BLU][COMMM]"
450 PRINTSPC(Z)"[SHIFTL][4COMM@][SHIFT@]
"
460 P=8:GOTO50
470 PRINTSPC(Z)"[SHIFTO][4COMMT][SHIFTP]
"
480 PRINTSPC(Z)"[COMMG][GRN][4COMM£]
[BLU][COMMM]"
490 PRINTSPC(Z)"[COMMG][GRN][RVSON][4COMM£]
[RVSOFF][BLU][COMMM]"
500 PRINTSPC(Z)"[SHIFTL][4COMM@][SHIFT@]
"
510 P=9:GOTO50
520 PRINTSPC(Z)"[SHIFTO][4COMMT][SHIFTP]
"
530 PRINTSPC(Z)"[COMMG][BLK]*[RED]*[CYN]
*[PUR]*[BLU][COMMM]"
540 PRINTSPC(Z)"[COMMG][YEL]*[GRN]*[PUR]
*[CYN]*[BLU][COMMM]"
550 PRINTSPC(Z)"[SHIFTL][4COMM@][SHIFT@]
"
560 P=10:GOTO50
562 E=INT((RND(1)*10)+1)
563 IFE=5ORE=10THEN565
564 GOTO600
565 PRINT"[HOM][11CRSRD][SPC][YEL][RVSON]
[3SPC][RVSOFF][SPC][RVSON][3SPC][RVSOFF]
[SPC][RVSON][3SPC][RVSOFF][SPC][RVSON]
[3SPC][RVSOFF][SPC][RVSON][3SPC][RVSOFF]
[BLU]"
566 PRINT"[SPC][YEL][RVSON][3SPC][RVSOFF]
[SPC][RVSON][3SPC][RVSOFF][SPC][RVSON]
[3SPC][RVSOFF][SPC][RVSON][3SPC][RVSOFF]
[SPC][RVSON][3SPC][RVSOFF][BLU]"
567 PRINT"[SPC][YEL][RVSON][3SPC][RVSOFF]
[SPC][RVSON][3SPC][RVSOFF][SPC][RVSON]
[3SPC][RVSOFF][SPC][RVSON][3SPC][RVSOFF]
[SPC][RVSON][3SPC][RVSOFF][BLU]"
569 I=INT((RND(1)*5)+1)
571 IFI=1THENPRINT"[2CRSRU][2CRSRR]1
";
573 IFI=2THENPRINT"[2CRSRU][6CRSRR]2
";
574 IFI=3THENPRINT"[2CRSRU][10CRSRR]
```

```
3";
575 IFI=4THENPRINT"[2CRSRU][14CRSRR]
4";
576 IFI=5THENPRINT"[2CRSRU][18CRSRR]
5";
577 Q=INT((RND(1)*20)+1)
578 IFQ=7THENPOKE36876,0:GOTO582
580 T=INT((RND(1)* 50)+187):POKE3687
6,T:FORG=1TO50:NEXTG:PRINT"[2CRSRL][YEL]
[RVSON][2SPC][RVSOFF][CRSRD][BLU]"
```

```
581 GOTO569
582 PRINT:PRINT"[2CRSRD][BLK]COLUMNA
[SPC]1,2,3,4[SPC]ACABAR";
583 IFI=0THEN600
584 GETV$:IFV$<>"2"ANDV$<>"1"ANDV$<>
"3"ANDV$<>"4"THEN584
585 IFV$="4"THENFORG=187TO231:POKE36
876,G:NEXTG:POKE36876,0:GOTO600
586 IFV$="1"THENZ=1:C=1:F=98:GOTO50
587 IFV$="2"THENZ=8:C=30:F=98:GOTO50
```

SOLUCIONES PESETAS

500	$ $ $ $	$ $ $ $	$ $ $ $
300	♥ ♥ ♥ ♥	♥ ♥ ♥ ♥	♥ ♥ ♥ ♥
450	* * * *	* * * *	* * * *
250	○	○	$ $ $ $
50	Λ	Λ	✕
50	✕	Λ	Λ
25	Λ	✕	✕
25	✕	✕	Λ

SOLUCIONES PESETAS

1.000	$ $ $ $	$ $ $ $	♥ ♥ ♥ ♥
500	♥ ♥ ♥ ♥	♥ ♥ ♥ ♥	✕
100	✕ 1	✕ 1	✕ 1
75	✕	✕ 1	✕ 1
75	✕ 1	✕ 1	✕

✕ UNA FIGURA CUALQUIERA

✕ 1 UNA FIGURA CUALQUIERA PERO QUE TIENE QUE SER IGUAL A LA QUE LLEVE ESTE MISMO SIGNO.

SALVAGUARDA DE MEMORIA EN PERIFERICOS

Rafael Jiménez Jiménez
I.B. "Nicolás Salmerón y Alonso"
C/Celia Viñas, s/n
ALMERIA

El programa descrito a continuación permite conservar sobre Diskette o Cassette el contenido de una serie de posiciones de memoria RAM del Commodore-64, especificada por el usuario mediante las direcciones inicial y final de la misma.

Tenemos abiertas así varias posibilidades; entre otras cabe citar las siguientes:

1.— Se podrán grabar programas en lenguaje máquina, sin necesidad de "enterrarlos" en sentencias DATA de un programa BASIC, con el consiguiente ahorro de memoria y tiempo de ejecución.
2.— Se podrá convertir el Diskette o el Cassette en una memoria "virtual" de pantalla de capacidad prácticamente ilimitada y, en el caso de utilizarse el Diskette, muy rápida.

Funcionamiento del programa

Al ser ejecutado es necesario introducir, en el orden en que son solicitados, los siguientes datos:

1.— Nombre que pondremos al fichero constituido por el trozo de memoria a grabar.
2.— Dispositivo de almacenamiento a utilizar. Se ingresará un "1" en caso de ser el Cassette o un "8" en caso de ser el Diskette.
3.— Dirección inicial (en decimal) del área de memoria a conservar.
4.— Dirección final (en decimal) del área de memoria a conservar.
5.— Autorización para iniciar el proceso de grabación.

Descripción del programa

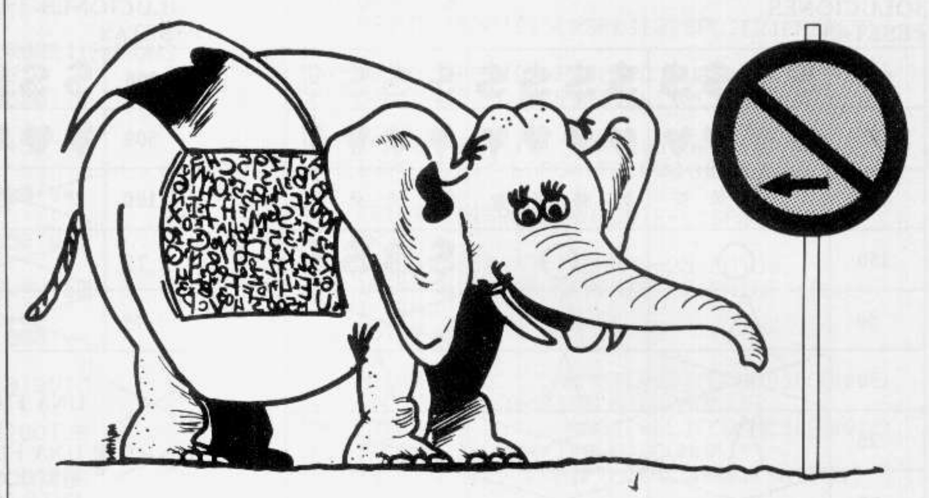
Consta de dos partes. Una de ellas en BASIC y la otra en código máquina.

La parte BASIC puede considerarse a su vez formada por otras dos. La primera está formada por la línea 10 y las sentencias DATA de las líneas 1000, 1010 y 1020, además de la subrutina de la línea 2000. Su cometido es cargar a partir de la posición de memoria $C000 la parte en código máquina.

El resto del programa BASIC sirve únicamente para recibir, preparar y colocar adecuadamente los datos que la mencionada subrutina en código máquina requiere.

Dicha subrutina, a su vez, lo único que hace es llamar a tres subrutinas del KERNAL y dispone los requerimientos necesarios para el funcionamiento de las mismas.

C-64



La más importante de las tres es la titulada SAVE. Su función es, precisamente, enviar a un periférico el contenido de un segmento de memoria. Su dirección de llamada es $FFD8 (dec. 65496).

SAVE no puede usarse sin más. Necesita "ayuda" y así, si se quiere poner un nombre al trozo de memoria a guardar, lo que usando el Diskette es imprescindible, hace falta el concurso de otras dos rutinas del KERNAL. Una de ellas es SETLFS. SETLFS prepara el número del fichero lógico, e indica el periférico a utilizar por SAVE y la dirección secundaria, "1" en nuestro caso. Esta dirección secundaria permitirá que en el momento de cargar al ordenador lo previamente grabado, se coloque en las mismas posiciones de memoria originalmente ocupadas. Su dirección de llamada es $FFBA.

La otra rutina KERNAL es SETNAM. Como su nombre indica tiene la misión de bautizar a la criatura, o dicho más seriamente, de darle un nombre al fichero que vamos a crear. Su dirección de llamada es $FFBD.

Veamos cuáles son los requerimientos de las rutinas anteriores y cómo se satisfacen. Empecemos por la primera en ser usada: SETLFS. Necesita que en el momento de su ejecución el acumulador del microprocesador contenga el número de registro lógico, que hemos fijado a "1". Esto se consigue con la primera instrucción LDA#01 (ver listado fig. 2). Otro requisito es que el registro X contenga el número de periférico, es decir un "8" para el Diskette o un "1" para el Cassette, de ello se encarga el primer POKE de la línea 80 (fig. 1), que como el 2º, el 5º y el 6º de la misma línea no hace sino introducir valores en los campos de operando oportunos del programa en lenguaje máquina. Finalmente es necesario que el registro Y contenga la dirección secundaria a utilizar, la "1", de lo cual se encarga la primera instrucción LDY#01 de la fig. 2. SETNAM necesita que en el momento de su ejecución el acumulador del micro contenga la longitud (nº de caracteres) del nombre del fichero, (2º POKE de la línea 80); el registro X contenga el byte menos significativo de la dirección a partir de la cual se encuentran los códigos ASCII de los caracteres del nombre y que el registro Y contenga el byte más significativo. Por ejemplo, supongamos que se nos ha ocurrido llamar FEDERICO al fichero. Pues bien, la sentencia 90 del programa BASIC pone los códigos ASCII de los caracteres de FEDERICO a partir de la dirección $C01C (dec 49178); por tanto en el acumulador introduciremos el valor 8, que es el LEN de FEDERICO, mediante el segundo LDA # del listado de la figura 2 y el segundo POKE de la línea 80; y en el registro X cargaremos $1C y en el Y $CO (LDX#1C y LDY#CO).

Después de todo esto estamos casi en condiciones de llamar a SAVE. Falta colocar la dirección inicial de la zona de memoria a grabar en dos direcciones consecutivas de página cero. El byte menos significativo en la primera y el más significativo en la siguiente. Aclaremos esto: la memoria del

ordenador está organizada en páginas, que no son otra cosa que series de 256 bytes consecutivos. La página cero consta de los 256 primeros, desde el $00 al $FF. Esta página está muy ocupada por el sistema operativo del 64; sin embargo, las direcciones $FB, $FC, $FD, $FE (del. 251, 252, 253, y 254) están libres. Utilizaremos la $FB y la $FC.

EN $FB colocaremos el byte menos significativo de la dirección inicial del área de memoria a grabar, de ello se encarga el tercer POKE de la línea 80. Y en $FC el byte más significativo, mediante el cuarto POKE de la mencionada línea.

El INPUT de la línea 60 y la subrutina 3000 se encargan del cálculo del valor de ambos bytes a partir del número decimal correspondiente.

El byte de página cero escogido, $FB, debe ser puesto en conocimiento de SAVE colocándolo en el acumulador. Es la función que cumple el último LDA≠FB del listado de la fig. 2.

Resta colocar en los registros X e Y la última dirección del área a grabar. El INPUT de la línea 70 y la subrutina 3000 junto con los últimos POKES de la 80 lo hacen.

Tras esto todo está listo para el comienzo de la grabación. La línea 100 no es más que una confirmación de seguridad. El SYS (49152) de la línea 110 lanza y completa la grabación, terminando el programa.

Carga en memoria del área grabada

Para leer de nuevo a RAM lo grabado se puede actuar en modo comando directo o desde un programa en BASIC. También puede hacerse desde una rutina en lenguaje máquina, pero acudiendo de nuevo al

FIGURA 2. Listado de la parte en Código Máquina.

Dir. Memoria (Hex)	C.M.	Ensamblador
C000	A9	LDA≠01
C001	01	
C002	A2	LDX≠N.P.
C003	N.P.	
C004	A0	LDY≠01
C005	01	
C006	20	JSR SETLFS
C007	BA	
C008	FF	
C009	A9	LDA≠N.L.
C00A	N.L.	
C00B	A2	LDX≠1C
C00C	1C	
C00D	A0	LDY≠C0
C00E	C0	
C00F	20	JSR SETNAM
C010	BD	
C011	FF	
C012	A2	LDX≠L.F.
C013	L.F.	
C014	A0	LDY≠M.F.
C015	M.F.	
C016	A9	LDA≠FB
C017	FB	
C018	20	JSR SAVE
C019	D8	
C01A	FF	
C01B	60	RTS

KERNAL del cual, por esta vez, ya hemos hecho uso más que suficiente.

La sintaxis para cargar a RAM en modo comando es:

LOAD "nombre de fichero", núm. de perif., 1

Ejemplo LOAD "FEDERICO" 8, 1 cargaría el archivo almacenado con el nombre de FEDERICO desde el Diskette.

El modo más simple a mi juicio, de cargar desde un programa en BASIC es colocando como primera línea de programa la siguiente:

10 IF A=0 THEN A=1: LOAD "nombre fich.", núm. per., 1

Conclusión

Resta indicar que el programa presentado puede modificarse dándole la forma de subrutina, con el fin de hacer que forme parte de otro mayor, o bien haciendo que los parámetros de entrada los genere el propio programa, etc. etc.,

Esta parte del trabajo es para el lector, que sabrá adaptarlo todo a sus propias necesidades.

```
10 FORI=49152TO49179:READA$:GOSUB200
0:POKEI,D:NEXT
40 PRINT"[CLR]":INPUT"NOMBRE";N$:N=L
EN(N$):IFN>100ORN<1THEN40
50 PRINT:INPUT"1-CASS[2SPC]8-DISK";P
:IFP<>8ANDP<>1THEN50
60 PRINT:INPUT"DIR.[SPC]INICIAL[SPC]
DE[SPC]MEMORIA";DY:GOSUB3000:LI=BB:M
I=BA
70 PRINT:INPUT"DIR.[SPC]FINAL[SPC]DE
[SPC]MEMORIA";DY:GOSUB3000:LF=BB:MF=
BA
80 POKE49155,P:POKE49162,N:POKE251,L
I:POKE252,MI:POKE49171,LF:POKE49173,
MF
90 FORI=1TON:POKE49179+I,ASC(MID$(N$
,I,1)):NEXT
100 INPUT"OK";C$:IFC$<>"SI"THENGOTO4
0
110 SYS(49152):STOP
1000 DATAA9,01,A2,00,A0,01,20,BA,FF
1010 DATAA9,00,A2,1C,A0,C0,20,BD,FF
1020 DATAA2,00,A0,00,A9,FB,20,D8,FF,
60
2000 D=0:FORJ=1TO2:D%=ASC(A$):D%=D%-
48+(D%>64)*7:A$=MID$(A$,2):D=16*D+D%
:NEXT
2010 RETURN
3000 BA=INT(DY/256):BB=DY-256*BA:RET
URN
```

fig. 1

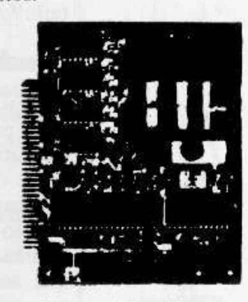
NOVEDAD

40/80 columnas para el VIC-20

La casa Ferre-Moret S.A. de Barcelona ha lanzado este mes al mercado español una tarjeta que permite convertir el VIC-20 en un ordenador personal de muy superior categoría.

Entre las características de esta tarjeta podemos mencionar que no ocupa memoria de usuario (posee su propia RAM de pantalla), no interfiere el funcionamiento normal del VIC-20, se puede alterar el espaciado entre las líneas de texto, el número de columnas por línea, proteger líneas de la pantalla para mantener mensajes fijos en ellas sin que les afecte el scroll y borrado de pantalla normales, y lo principal... se ven caracteres muy bien definidos y nítidos. Quizás sea ésta una de las características más decisivas a la hora de utilizarlo conectado a un gran ordenador (ver Noticias Commodorianas, pág. 6).

El precio de venta al público es de 17.250 pesetas y en mi opinión es uno de los accesorios a tener en cuenta si se desea utilizar el VIC-20 para algo más que matar marcianitos.



La tarjeta puede pedirse a:
Ferre-Moret S.A.
Tel.: 93-250 84 40
C/ Buenos Aires, 30, 2º, 3.ª
BARCELONA-36
Precio: 17.250 ptas.

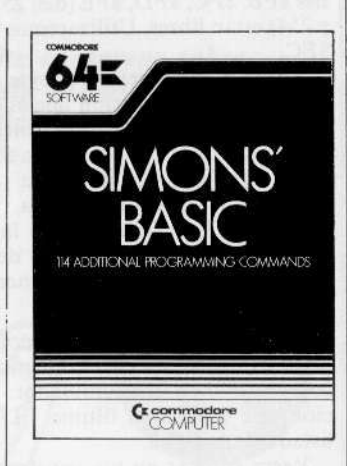


DE TOMO Y LOMO

NOVEDAD

SIMON'S BASIC

En INFORMAT tuvimos la oportunidad de conocer una de las últimas novedades de Microelectrónica y Control para el C-64, se trata del cartucho "SIMON'S BASIC", este cartucho añade 114 comandos a los que ya posee el C-64 como ya adelantamos en el número anterior.



Estos comandos los podemos dividir en varios tipos:

— Comandos de ayuda a la programación (KEY y TRACE).
— De manejo de cadenas (INSERT y PLACE).
— De texto (CENTRE y PRINT AT).
— De entrada de datos (FETCH e INKEY).
— Aritméticos (MOD y DIV).
— De conversión numérica a binario y hexadecimal.
— De programación estructurada (POC e I F...THEN...ELSE).
— De manejo de pantalla (SCRSV y COPY).
— Gráficos (CIRCLE y PAINT).
— De sprites y gráficos definidos por el usuario.
— De música (WAVE y ENVELOPE).
— De operación con discos (DIR y DISK).

He enumerado algunos de los comandos como ejemplo, pero la lista de comandos es enorme (114), son todos aquellos que los usuarios estaban esperando para los distintos campos de aplicaciones de su ordenador personal, los aficionados a las matemáticas y dibujos técnicos querían un cartucho que les añadiese los comandos PLOT, LINE, etc., los aficionados a programas de marcianitos esperaban un programa que permitiese definir sus propicos caracteres, los programadores novatos y los empedernidos del teclado esperaban algo que les ayudase a detectar errores y probar programas, etc... Este cartucho es la respuesta para todos.

Aspectos positivos del Simon's Basic, podríamos seguir enumerando durante varias páginas de la revista y necesitaríamos varios números para publicar ejemplos donde demostrar las cualidades, pero no vamos a hacerlo.

Como todo debemos decirlo, vamos con los aspectos negativos: en primer lugar el precio que lo aleja del alcance de los jóvenes aficionadillos que tienen pocas cucas. El segundo inconveniente es el manual (viene en inglés de momento), pero pronto lo tendrán en español. El tercero es que aquellos programas que utilizan variables de más de dos caracteres no funcionan, da SYNTAX ERROR en las líneas que tengan variables más largas. Esto último se soluciona quitando el resto del nombre de la variable ya que aunque el basic de Commodore las acepta, sólo lee los dos primeros caracteres.

El precio del cartucho SIMON'S BASIC es 16.000 pesetas, comercializado por MEC.

Algunos programas de uso común en BASIC (edición para PET/CBM)

Autores:
Lon Poole
Mary Borchers
Carroll Donahue

Editorial:
Mc.Graw-Hill España.
Sta. Beatriz, 4.
Madrid-18
P.V.P. 1.900 pesetas

Nos encontramos ante un libro dedicado a los PET/CBM. El enfoque dado ha sido práctico, los autores lo advierten en la introducción, no han pretendido realizar un curso de Basic, sino recopilar 76 programas que pueden resultar positivos para el aprendizaje al analizarlos.

Hemos probado varios de los programas que aparecen listados y funcionaron a la primera. La temática se centra en rentabilidad, amortización, préstamos, integración, derivación, regresiones, estadística, etc.

Cada programa propuesto aparece con una pequeña introducción que expone el problema que queremos resolver, nos propone una fórmula matemática para su resolución, el significado de cada uno de los términos que aparecen en la fórmula unos ejemplos con los resultados que nos darían si desarrollamos correctamente el problema y finalmente aparece el listado del programa que corresponde a la fórmula propuesta.



Cada listado está repleto de sentencias REM para explicar lo que hace esa sección del programa, pero los comentarios de estas sentencias vienen en inglés. El resto del libro está correctamente traducido al español.

Este libro puede ser útil para todos aquellos que deseen utilizar el ordenador para la resolución de problemas matemáticos o aplicarlo a los negocios bancarios.

Casi todos los programas funcionan en el VIC-20 y C-64 sin tener que hacerles modificaciones, pero algunos pueden necesitarlas para conseguir representar las gráficas que originalmente eran para 40 columnas a la pantalla de 22 del VIC.

Magia



La MAGIA son trucos, la MAGIA es divertida.

La MAGIA es hacer lo que nadie se ha atrevido y resulta ser la fuente más completa de información para la informática práctica.

La MAGIA es una sección mensual llena de consejos, trucos, de esto y aquello del mundo del software, hardware y aplicaciones, trucos descubiertos por los demás que hacen que la informática sea más fácil, más divertida o más animada.

MAGIA habla de ideas sencillas, programas de una sola línea, subrutinas útiles, hechos de informática poco conocidos y otras cosas de interés.

PREMIOS: *(VER PAG. 6)*

Desconecta sin apagar

Hola queridos colegas de la COMMODORE WORLD, aquí os envio un pequeño truco para vuestro VIC.

Con este programa se obliga al ordenador a despreciar una cantidad de memoria y hacerle creer que no tiene conectada ninguna ampliación de 8 ó 16 K, programa muy útil para cargar programas que no requieran ampliaciones (STANDARD) y olvidarse si hay o no conectada ampliación alguna, sobre todo de más de 3 K, espero os guste, un saludo.

```
10 POKE36869,240:POKE36866,150:POKE 648,30
20 FORJ=217TO228:POKEJ,158:NEXT
30 FORJ=229TO250:POKEJ,159:NEXT
50 POKE56,30:CLR:POKE4096,0:POKE44,16:POKE198,2:POKE631,
   147:POKE632,131:NEW
```

Diego Herrero Ortega
Avenida Jaime I, 317, 4º 2.ª TERRASSA (Barcelona)

Strings Nulos y ASC

Cuando al ejecutar un programa, salga ?ILLEGAL QUANTITY en una función ASC, sólo tenemos que sumar CHR$ (Ø) al final de la cadena y el problema se soluciona. Ejemplo:

```
ERROR      10 Z=ASC(Z$)
SOLUCION   10 Z=ASC(Z$ + CHR$ (Ø))
```

Os doy las gracias por leerlo. Sólo os quiero preguntar algo. Cuando recibí el envío de vuestras tres revistas, encontré un número entre paréntesis al lado de mi dirección. El nº es 4884, quisiera saber si es mi número de suscripción o el nº de envio, o algo así. Un abrazo a todos y espero que vuestra adolescente dure muchos años, por el bien de todos.

Isabel Moreno
R/ Pujolet, 10-1º-1.ª. BAÑERAS (Tarragona).

Barra de tiempo

El presente programa crea una barra de tiempo, la cual puede ser útil para algún programa de juegos.

```
 10 REM Barra de tiempo
 20 E$="(CTROL 2) (RVSON) (30SPC)"
 30 PRITE$
 40 N=30
 50 N=N-1
 60 PRINT"[CRSRU]"TAB(N);CHR$(32)
 70 IF N=O THEN 100
 80 FORT=1 TO 1500: NEXT T
 90 GOTO 50
100 PRINT "TIEMPO AGOTADO"
110 END
```

El tiempo se puede modificar cambiando la línea 80 y el color y la longitud de la barra en la línea 20 junto con la línea, 40.

José Fco. Torres Caro
Apolo, 91. TORREVIEJA (Alicante)

VIC - Piano

Soy suscriptor vuestro desde el primer número y me parece bastante buena la revista, superior a Club Commodore. Quiero enviaros un pequeño programa, que si queréis incluirlo en Magia, es interesante para los "Músico Vic-viciosos. En resumen, hace que todas las teclas del "qwerty" o teclado se conviertan en un piano y no así las teclas de control, bueno, ahí va:

```
1 FORI=ØTO44
2 READA: POKE672+I,A
3 NEXT:SYS672:NEW
4 DATA 12Ø,169,176,141,2Ø,3,169,2
5 DATA141,21,3,88,96,Ø,Ø,Ø
6 DATA165,197,2Ø1,64,24Ø,12,9,192
7 DATA141,12,144,169,15,141,14,144
8 DATA2Ø8,8,173,14,144,24Ø,3,2Ø6
9 DATA14, 144,76,191,234
```

Este programa funciona incluso si se escribe luego NEW, y se efectúan otros programas y se recupera escribiendo SYS 672, no funciona si se apaga de Vic-20 claro!

Luis Solórzano Vázquez
Bartrina, 1-3, esc. B 3º-1.ª. BARCELONA-30

Freno para listados

Amigos de Commodore:

Esto es bastante útil para listar programas largos porque los lista línea a línea.

Se escribe list, e inmediatamente después de dar a la tecla "RETURN" se da a la tecla "CONTROL". Mientras se mantenga apretada esta última tecla el programa aparecerá línea a línea.

Espero que sirva de algo a los lectores. Hasta pronto.

Rafael Olalla Martínez
Eugenio Sellés, nº 3 - 5º B. PEDREGALEJO - Málaga

Cosillas en la pantalla

Os mando unos POKES por si merecen la pena publicarlos en la sección de MAGIA. Están hechos en un VIC-20.

POKE36867,32 hace pequeña la pantalla.

POKE36867,56 pone la pantalla normal, pero con unos signos al final de ella.

POKE34,7:POKE23,PEEK(43):SYS54386 borra la pantalla.

POKE36867,23 saca unos signos en pantalla y al final de ellos sale una A centelleante, pulsen teclas y verán lo que pasa.

Agustín Horrillo
Emilio Serrano, nº 1-2ºB. ERMUA (Vizcaya)

Encolumnado de Números

El ordenador presenta las cantidades ajustadas por la izquierda, con lo que quedan de forma incorrecta, ya que las unidades, decenas, centenas, etcétera, no ocupan la misma columna.

El siguiente programa obvia este inconveniente

```
100 INPUT A: A$= STR$(INT(A)): LO= LEN(A$):
    PRINT TAB(20 - LO)A: GOTO 100
```

Manuel Hernández Báez
Palencia, 29, Madrid-20

Input sofisticado

Se basa en un INPUT más sofisticado, más profesional. Nosotros cuando desde cualquier parte del programa, queramos hacer un INPUT de este tipo, sólo hemos de darle el número de caracteres que se entrarán (LG=nº de caracteres) y hacer actuar la subrutina. Luego ponemos N$=G$ (pongo n$ como ejemplo) y ya hemos utilizado nuestro INPUT sofisticado.

```
1ØØØ FORI=1TOLG:G$=G$ "[spc]":NEXTI
1Ø1Ø P=P+1:IFP>LGTHENP=LG
1Ø2Ø IFP<1THENP=1
1Ø3Ø PRINT"[CRSRU]"LEFT$(G$,P-1)"[RVSON]"MID$(G$,P,1)
     "[RVSOF]"RIGHT$(G$,LG-P)
1Ø4Ø GETA$:IFA$=""THEN1Ø4Ø
1Ø5Ø IFA$=CHR$(13)THENPRINT"[CRSRU][RVSON]"G$"[RVSOF]
     ":RETURN
1Ø6Ø IFA$="[CRSRD]"ORA$="[CRSRU]"ORA$=CHR(2Ø)ORA$=
     CHR$(148)THEN 1Ø4Ø
1Ø7Ø IFA$="[CRSRL]"THENP=P-2:GOTO1Ø1Ø
1Ø8Ø IFA$="[CRSRR]"THEN1Ø1Ø
1Ø9Ø G$=LEFT$(G$,P-1)+A$+RIGHT$(G$,LG-P)
11ØØ GOTO1Ø1Ø
```

En el caso de lo que queramos sea una variable numérica entonces, hemos de modificar en la subrutina lo siguiente:

```
1Ø41 IFA$="."THENGØTØ1Ø9Ø
1081 IFASC(A$)  480RASC(A$)  57THEN1Ø4Ø
1Ø5Ø IFA$=CHR$(13)THENPRINT"[CRSRU][RVSON]"G$"[RVSOF
]":G=VAL(G$)
RETURN.
```

Hasta pronto amigos.

Santiago Casas Duarte (16 años)
Emp. Eugenia, 14 - 1º. TARRASA

Gráficas en Baja Resolución

Amigos de Commodore World, os envío la siguiente colaboración para el apartado de MAGIA a la vez que os felicito por vuestra estupenda revista.

Para todos los principiantes que como yo nos cuesta fabricar estupendas figuras en la pantalla del C-64 a base de sprites y programas largos, aquí está este original programa que, aunque corto, produce una figura gráfica realmente magnífica.

Hacedlo con el cursor en negro, intentando también sustituir el 86 de la línea 5Ø por 91,97,98,114,1Ø2, ó el que queráis.

La línea 7Ø coloca mi nombre en el centro de la figura; no soy vanidoso, así que podéis colocar cada cual el vuestro.

Hasta pronto.

```
1Ø PRINT "[CLR]"
2Ø POKE 532ØØ,5 : POKE 53281,7
3Ø FOR T = Ø.1 TO 6.2 STEP Ø.Ø2
4Ø X=INT(12*COS(T) ↑ 3) : Y=INT(12*SIN(T) ↑ 3)
5Ø POKE 1563+X+4Ø*Y,86
6Ø NEXT
7Ø PRINT "[HOME]" ; "[16CRSRR] [12CRSRD]" ; "MANOLO"
8Ø GOTO 8Ø
```

Manuel Veamonte Serrano
Depósitos Bajos, S/N. CASETAS (Zaragoza)

Metedura de pata en "números primos"

El motivo de la presente es para indicarles que existe un error de imprenta en la rutina para generar números primos que les envié y apareció publicada en el número de mayo en la sección MAGIA

En concreto se trata de una omisión. En la página 54 aparece escrita la línea tres de la subrutina, asignando C = I .5 cuando en realidad debería de haberse imprimido en la forma siguiente: C = I ↑ .5 es decir, se omitió el signo de potenciación (↑). Ruego rectifiquen este error en la sección de METEDURAS DE PATAS del próximo número.

Rafael García Segura
Montes de Oca, 9-2º. Málaga-7

Paradas en programa

Estoy sorprendido de que siempre sale un "truco" sobre la orden:

```
10 GET A& : IF A& = "x" THEN (lo que sea)...
```

Si esta orden es tan sólo para detener un programa esto también se puede realizar mediante una orden INPUT. Me explico: si en el transcurso del programa colocamos un INPUT, éste se detendrá y aparecerá un interrogante en pantalla; tan sólo apretando el RETURN el programa continuará, fácil, ¿no?!.

Santiago Martínez Romero

Ecuaciones, Barquitos y Manual

Como pedís programas de una sola línea para el apartado "Magia" he decidido mandaros uno, para que lo publiquéis si os gusta.

El programa resuelve ecuaciones de segundo grado.

Funcionamiento:

El ordenador pide los coeficientes de la fórmula general A,B,C,

Halla los valores de X(Z e Y en el programa)

Escribe estos valores

Vuelve a empezar (para resolver otra ecuación)

La instrucción Print hay que ponerla abreviada (?) para que quepa en una línea.

P R O G R A M A

```
1 INPUTA,B,C:Z=(—B+SQR(B↑2—4*A*C))/(2*A):Y=(—B—SQR(B↑2
—4*A*C))/(2*A):?Z,Y:GOTO1
```

En el programa Batalla naval que publicabais en el nº 2 (marzo), cuando el ordenador da en uno de los barcos del jugador, puede repetir la jugada, lo que no ocurre al revés, cuando el jugador acierta.

Insertando una línea, este problema se soluciona. La línea es:

```
575 GOTO 450
```

(es válido para el V-20 y el C-64)

¡Ah! y el manual del usuario del C-64 no tiene sólo el par de errores que publicabais sino unos cuantos más.

Jorge Lozano del Amo (16 años)
Subida de S. Andrés, nº 1. AINZON (Zaragoza)

NOTA: ENVIANOS TODOS LOS ERRORES QUE ENCUENTRES.

Mensajes en Sentencias Input

Queridos amigos:

En el apartado de MAGIA de la revista de marzo habéis publicado unas líneas que os envié. Me alegro mucho que haya sido útil a otros lectores de la revista. Como no había enviado la dirección lo hago ahora y de paso os envío otra observación con respecto al VIC-20. Va dirigida a los que llevan poco tiempo con el ordenador ya que el resto supongo que ya la conocerá:

Cuando se mete un mensaje entrecomillado después de la instrucción INPUT ha de tenerse en cuenta la longitud de dicho mensaje pues no debe sobrepasar un número determinado de espacios, dependiendo del ordenador que sea, en concreto para el VIC-20 admite hasta 20 espacios o pulsaciones. En caso de no conocer este dato si se ponen caracxteres de más el ordenador repite la misma línea pidiendo datos y con un mensaje de error sin poder llevar el programa adelante.

Andrés Pozo
Otxarkoaga Bl. 22-N7-6ºD. Bilbao-4

Rutina Move Memory

Hola amigos de COMMODORE:

Aquí os envío una subrutina que sirve para copiar una zona de memoria (dirección fuente) en otra (dirección destino).

Haciendo uno de ella podremos obtener una copia en RAM del BASIC o juego de CARACTERES, por poner un ejemplo, en un tiempo mínimo (menos de un segundo). Esta subrutina llama a una rutina del BASIC, en l.m., que se llama MOVE MEMORY y está en la dirección $A3B8.

Antes de recurrir a ella debemos cargar los punteros necesarios en las siguientes direcciones: (esta operación la hacen las líneas: 1000, 1010, 1020 y 1030)

DIRECCION 90-91...PUNTERO DE DIRECCION FUENTE
DIRECCION 88-89...PUNTERO DE DIRECCION DESTINO
DIRECCION 95-96...PUNTERO DE NUMERO DE BYTES A TRASLADAR.

Los parámetros de entrada en la subrutina son:

VARIABLE F1 -- DIRECCION FUENTE
VARIABLE D1 -- DIRECCION DESTINO
VARIABLE N1 -- NUMERO DE BYTES A TRASLADAR

LISTADO DE LA SUBRUTINA

```
1000 F1=N1+F1- 1 : D1=D1+N1- 1
1010 H1% = F1/256 : L1 = F1 - H1%*256
1020 H2% = D1/256 : L2 = D1 - H2%*256
1030 H3% = N1/256 : L3 = N1 - H3%*256
1040 POKE 90,L1 : POKE 91,H1%
1050 POKE 88,L2 : POKE 89,H2%
1060 POKE 95,L3 : POKE 96,H3%
1070 SYS 41919 : RETURN
```

Se puede simplificar más el número de líneas de BASIC, de esta rutina, haciendo previamente los cálculos que realizan las líneas 1000, 1010, 1020, 1030 y poner en las líneas 1040, 1050 y 1060 los valores obtenidos de las variables correspondientes.

Recibid un saludo y una felicitación por la revista que ha mejorado mucho.

J. R. L. (Iniciales por expreso deseo)
BERGARA (Guipúzcoa)



CARTA BLAN

RESTORE SORDO



Me gustaría que me aclarasen una duda: Para devolver el C-64 a su presentación normal (después de haber usado colores, inversión, etc.) se pulsa primero RUN/STOP, y sin soltar esta, se aprieta RESTORE. Pero en mi C-64 casi nunca me funciona la primera vez, tengo que repetirlo varias veces hasta que me salga READY. Sé que no soy el único que me pasa esto, y me gustaría saber a qué es debido.

XAVIER PRAT I BURDO
C/ RABAL STA. EULALIA S/N.
STA. MARIA D'OLO (BAGES)
BARCELONA

Esto se debe a las rutinas del sistema operativo (KERNAL) del C-64 y es común a todos los C-64. No le pasa nada a tu aparato.

¡PUES NO JUEGO, HALA!



El programa "Batalla Naval VIC-20" (Commodore World, nº 2, pág. 36) es inaceptablemente injusto para con el jugador humano.

Todo queda equilibrado añadiendo: 585 IF PE > 57 AND PE < 63 THEN 450.

En caso contrario, sólo el enemigo tiene derecho a repetir tras un acierto, y... ¡así, cualquiera!

GUILLERMO FATAS
STRA. TERESA, 58
ZARAGOZA-6

Vale hombre, que ganes muchas partidas.

UNA DE INDIOS



Me gustaría saber qué función tiene la tecla superior izquierda del VIC-20 que es una flecha horizontal que apunta hacia la izquierda.

No tiene función especial, es como cualquier otra (ejemplo; asterisco). Sólo en el superexpander que activa el sonido. "El Romance con el VIC" (Comm. World Nº 3) la utiliza.

2 cartas contestadas por MEC

PASCAL PARA C-64



Poseo un C-64, estoy muy contento con la compra del aparato y agradezco el esfuerzo de la revista; pero mi decisión de la compra se consolidó al aparecer desde mayo a navidades del 83, la aparición en un futuro inmediato del cartucho PASCAL U.C.S.D.

En la nueva propaganda que presenta Microelectrónica ya ni figura relación a posible a dicho lenguaje.

Agradecería una contestación elocuente. Y en el caso de ser una mentira por causas de estrategia comercial, desearía saber si está previsto y el tiempo en aparecer en el mercado de un lenguaje en programación estructurada, que es la finalidad que perseguí en mi compra.

J. F. FERNANDEZ AREVALO
AVDA. MEDITERRANEO, 39-3-1
MADRID-7

SEGUIMOS CON PASCAL



Soy un suscriptor de su revista "COMMODORE WORLD" y les escribo para hacerles una consulta.

Esta viene dada al recibir la noticia de que el ordenador Sinclair Spectrum tiene un programa compilador PASCAL en 12 K, que posteriormente he comprobado su correcto funcionamiento. La consulta es si está previsto o está en estudio algún programa similar en cassette o cartucho de PASCAL, o en algún otro lenguaje de programación (p.e. PROLOG), para el VIC-20. Igualmente si es afirmativo me gustaría saber en cuánto tiempo está previsto que saldría al mercado.

Me interesa mucho su respuesta y me gustaría poder trabajar con él en mi ordenador personal.

Me despido no sin antes agradecerles su atención, y esperando una pronta respuesta.

ANTONIO LUQUE LUZ
C/ FRANCIA, 56-1º, 2.ª
HOSPITALET. (BARCELONA)

Lamentablemente no es el suyo el único caso del que tenemos noticias. Nosotros no sabemos aún de la existencia de ningún PASCAL en el mercado para el Commodore. De todas formas, Commodore prometió que sacaría uno. Esa es la razón por la cual nosotros lo incluimos en la propaganda del 64. Y seguimos incluyéndolo. Esto puede comprobarlo en el folleto que le adjuntamos. La razón es que estamos convencidos de que Commodore lo sacará, ya que también prometió el LOGO, y lo hizo.

De todas formas nosotros estamos haciendo lo imposible intentando conseguir algún PASCAL por alguna otra fuente, ya que lo seguimos considerando como una interesante opción para el Commodore 64.

Por las razones expuestas anteriormente comprenderá que no le podamos dar un plazo de aparición en el mercado nacional, aunque intentaremos con todos los medios a nuestro alcance que este plazo sea lo más breve posible.

Nos es grato comunicarle la existencia de otros lenguajes estructurados ya disponibles para el Commodore 64, tales como el SIMON'S BASIC, el LOGO y el FORTH.

MICROELECTRONICA Y CONTROL, S.A.
DPTO. ORDENADORES PERSONALES.

LO DICE EL MANUAL



1º. Cómo podría efectuar el proceso contrario de PEEK, o sea un número convertido en su patrón de bits pasarlo a su estado normal por llamarlo de alguna forma.

2º Cómo hacer que un programa, de juegos por ejemplo, se realicen varias cosas a la vez, es decir, que tú estés disparando y al mismo tiempo se estén moviendo los disparados.

3º Cómo se pueden realizar dibujos complicados como por ejemplo un avión.

4º Por último me gustaría saber para qué se puede aplicar el código ASC.

Antes de despedirme quiero felicitarles por esta maravillosa revista.

Ah! tento 14 años

JAVIER TORRES DE ZUBIZARRETA
C/ RAMBLA DE MENDEZ NUÑEZ, 45
ALICANTE-2

1º. El proceso contrario al de PEEK lo realizas con el comando POKE, y viene explicado en el manual de tu programador.

2º. Simplemente diseñándolo así en el programa. En lugar de hacer que el programa mueva el disparo hasta el final, haz que lo mueva un paso, comprueba si has vuelto a disparar, mueve los ovnis, y vuelve a mover el primer disparo, así hasta el final.

3º. Los dibujos algo complicados requieren la ayuda del cartucho superexpander y/o un programa ingenioso que los defina y permita utilizarlos, mira el programa "Aventuras en el desierto" en el número 4 de la revista (el jeep está definido como varios caracteres juntos).

4º. El American Standard Code for Information Interchange, es el código utilizado en casi todos los ordenadores modernos,

CA... Y...

y la función ASC en Basic nos devuelve al código ASCII del "string" o cadena que utilicemos como argumento, te puede servir para clasificar, comparar strings, etc.

EL SIMONS LO HACE

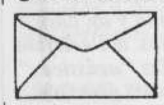


Quisiera saber si existe para el C-64 un cartucho como el que existe para el VIC-20, que se llama SUPEREXPANDER y que tiene por objeto interceptar el Basic de Commodore incrementando sus instrucciones y comandos en gráficos y sonidos, introduciendo comandos como GRAPHIC, COLOR, POINT, DRAW, CIRCLE, PAINT, SOUND, etc.

MAURICIO JAVIER M. GONZALEZ
PJE. RIBATALLADA, 11-3º, 1.ª
SABADELL (Barcelona)

Ya existe un cartucho similar al superexpander para el C-64, se trata del SIMON'S BASIC, lo comercializa Microelectrónica y Control, y la reseña de este cartucho sale en este mismo número.

¿CONFUSION DE PATILLAS?



En el nº 3 de Commodore World, apareció un artículo titulado "Cómo añadir una tecla RESET". Pues bien el artículo me pareció bien y fabriqué dicha tecla, la conecté y efectivamente funcionaba perfectamente. Pero cual es mi sorpresa que al volver a conectar la impresora, esta, ya no recibía las instrucciones del C-64. Más exactamente, para listar un programa por la impresora pongo OPEN 1,4 : CMD 1 : LIST y pulso RETURN, no ocurre nada, es decir la impresora ni se inmuta y en la pantalla sigue la instrucción tal cual, como si esperara que la impresora terminase. Si la impresora no está presente el ordenador lo detecta.

Tengo que decir también que la impresora debe estar bien pues la he comprobado con un VIC-20 y funciona perfectamente.

Es por lo que ruego me contesten a las siguientes preguntas:

1º ¿Puede haberse estropeado la CPU?

2º ¿Puede haber sido a causa de la tecla Reset?

3º ¿Si no, qué puede haber ocurrido?

4º ¿Qué es lo que puedo hacer "yo" aparte de llevarlo a arreglar?

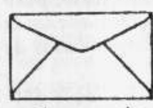
1º. La CPU no puede estropearse con el pulsador de RESET.

2º. El fallo no puede ser causado por la incorporación de esta tecla si las conexiones fueron correctas.

3º. Puede haberse estropeado la CIA (6526) que se encarga de las comunicaciones con periféricos a través del bus serie, esto pudo ser causado por un error en el conexionado o por desconectar el cable del bus serie con la tensión aplicada al 64 y/o la impresora.

4º. Puedes enviarnos tus señas y teléfono, en la carta no aparecían.

ORDENADOR PERSONAL O CRONOMETRO DE PRECISION



Quisiera haceros una pregunta sobre la función TI$. Dicha función activa el reloj interno del ordenador y emite la hora en tiempo real. En un caso he comprobado que dicha hora se atrasa 1 segundo cada 4 horas, o sea, 6 segundos al día, con mi C-64. Esto sin realizar ninguna operación de E/S con ningún periférico, pues ya sé que el reloj se detiene si utilizo el Datassete.

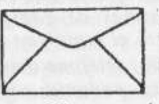
Quisiera saber si esto es normal en todos los ordenadores o simplemente es un defecto del mío.

JORDI GONZALEZ MIRABET
AV. BORBON, 72
BARCELONA-16

El ordenador no es un cronómetro de precisión, si quieres corregir este error de cuatro segundos diarios deberás llevarlo al laboratorio donde te lo calibrasen con un Patrón de Precisión. Pero creo que sería pasarse un poco ya que para mantener esa precisión deberías mantener la temperatura constante, si la temperatura varía lo hace el ajuste (dentro de unos límites).

Es perfectamente normal ese "retraso" en tu reloj.

PASCAL Y TRES DIMENSIONES



Soy suscriptor de la revista "Commodore World" y les escribo para preguntar una serie de cosas:

—Me gustaría saber si existe (pienso que sí) algún cartucho o cinta para poder programar en pascal con el C-64, es el ordenador que tengo. De ser así, desearía recibir toda la información sobre ello (precio, dónde encontrarlo o pedirlo, etc.).

—También me gustaría que publicaran en su revista, algún problema sobre gráficos y dibujos tridimensionales (me gustaría saber cómo hacer estos dibujos 3-d, podrían escribir algún artículo sobre ello).

—En una librería pude observar la existencia de un libro dedicado íntegramente al C-64, pero en inglés, me gustaría saber el título, precio y dónde poder encontrar (o al menos la editorial para poderlos pedir) libros sobre el C-64.

JUAN ANTONIO OREJA MORENO
C/ ESTADIUM, 10-4º E
BADAJOZ

Tu primera pregunta creo que la aclara la contestación de Microelectrónica y Control a J.F. Fernández en este mismo número.

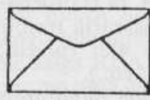
Intentaremos atender tu petición sobre dibujos en tres dimensiones, pero lo imposible lo hacemos en el acto, con los milagros tardaremos un poco más.

Quiero decir que no podemos atender a todos a la vez como podrás imaginar, pero danos tiempo y veras...

...SEAMOS PREGU

Libros en inglés sobre el C-64 existen en cantidades, los tienes sobre gráficos, música, programas listados, sus rutinas de código máquina, etc. No sé cuál será el tema que te interese, pero si nos llamas por teléfono o nos escribes especificándolo quizás podamos contestarte.

QUIERO UNA DE CATORCE



Poseo desde hace poco un C-64 y quisiera aclarar unas dudas y haceros al mismo tiempo una petición.

1º. ¿Es posible simultanear mayúsculas y minúsculas en la pantalla? ¿Cómo?

2º La tecla superior-izquierda ( ), ¿tiene alguna función especial?

3º ¿Podéis publicar algún programa de combinatoria aplicada a quinielas?

Por ejemplo, que mostrara las combinaciones posibles para un determinado número de 1, X, 2.

Agradeciéndoos de antemano vuestra atención, recibid un cordial saludo.

P.S. ¿Conocéis algún programa aplicado a la medicina para el C-64?

MIGUEL BARNOST BUFI
PADILLA, 282
BARCELONA-25

Para simultanear mayúsculas y minúsculas, pulsa las teclas SHIFT y Commodore (abajo a la izquierda) y ya puedes teclear.

La tecla no tiene ninguna función especial, el VIC la usa para activar el sonido con el super-expander, y en el artículo "Romance con el Vic" apareció. En realidad es otra tecla más en el teclado y tú la puedes utilizar como quieras en tus programas.

No es la primera vez que nos preguntan sobre programas de quinielas, pero no los tenemos, si nos llega alguno lo publicaremos.

Programas de medicina sabemos de los que hicimos la reseña en el número anterior (Metisistemas) y en uno de los próximos números aparecerá una reseña sobre otro de hipnosis, relajación, adelgazamiento, etc. Parece interesante...

QUIERO JUGAR CON VENTAJA



Les dirijo esta carta para saber si en el programa "Batalla Naval C-64" que publicaron en el nº 2 de su revista hay algún error, o simplemente falta alguna línea.

Yo tecleo cuidadosamente el programa (y lo comprobé dos veces) y resulta que cuando disparo a los barcos del ordenador, éste no me indica si lo he alcanzado o no, y en el caso de haberlo alcanzado no me permite volver a tirar, cosa que sí hace el ordenador cuando dispara contra mis barcos.

Dándoles las gracias por adelantado y en espera de su grata contestación, se despide de ustedes.

FRANCESC LLENAS I TORRENT
L'AIGÜETA, 166
LA BISBAL D'EMPORDA (GERONA)

No hay ningún error en la batalla naval del número 2. Cuando disparas suena el agua o el sonido de alcanzado y en su caso el de hundido. Si le das no tienes oportunidad de repetir ya que tú tienes lógica, eres inteligente, y él jugaría con desventaja al estar sus disparos realizados en modo aleatorio, pero si quieres igualar las cosas puedes añadir la línea que nos ha enviado Guillermo Fatas de Zaragoza:

585 IF PE>57 AND PE<63 THEN 450.

Ahora podrás repetir el disparo cuando aciertes, pero eres injusto con el ordenador..

INCOMPATIBILIDADES



¿Son compatibles los programas de CBM-4000 y 8000 con el CBM-3000 y los de V-20 y C-64 con CBM-3000, bien entendido que en este último modelo no operarían los comandos correspondientes a los anteriores modelos, como color, sonido, etc.

Al CBM-3032, que es el modelo que tengo, le he realizado unas conexiones con un radio-cassette (para utilizarlo solamente como altavoz) y he conseguido reproducir efectos sonoros de un programa, que puede copiarse en otro diskette por lo que entiendo que no está protegido, pero como me parece que está escrito en lenguaje máquina, no tengo ningún conocimiento del tema, no he sido capaz de listarlo en impresora, ni por pantalla claro, para aprender a utilizar los interiores de la CPU que posibilitan su implantación en otros programas, tanto recreativos, como para enviar señales de aviso en las aplicaciones que suelo utilizar. ¿Puede decirme alguien cómo puedo desbloquear la CPU para que me liste? (repito que el programa puede copiarse cuantas veces quiera o sea que no está protegido) ¿Commodore World podría enseñarnos a los usuarios del 3000 algunos trucos del lenguaje máquina que posibiliten efectos sonoros o hay alguien que lo sepa y quiera enseñarme?

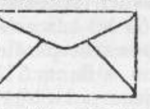
MIGUEL FERNANDO GARCES TERCERO
C/ SAN PEDRO, 2-8º A. EDIF. ANTICA
APARTADO DE CORREOS 286
ALMERIA

Los programas de otras series de Commodore son compatibles con el CBM-3000 dentro de unos límites, aparte de los comandos de color y sonido incorporados a modelos más modernos tienes que tener en cuenta las diferencias de las versiones de BASIC, enumerar las diferencias ocuparía todo un libro, y te puedo recomendar que des una ojeada a la "Guía del Ordenador Personal PET/CBM" McGraw Hill España.

Imagino que no tendrás un desensamblador, si lo tuvieses seguramente habrías listado el programa, pero de poco te servirá si no sabes código máquina, incluso sabiendo es difícil averiguar lo que hace cada sección del programa, sería necesario conseguir el listado del código fuante comentado para poder entenderlo... o una gran dosis de intuición...

Todo lo que nos manden o consigamos sobre el 3000 lo iremos publicando, pero no podemos probarlo ya que sólo tenemos en la redacción VIC-20, C-64 y C-700.

MODEM MODERNO CON GUITARRA



Hola, sólo quiero hacer un par de preguntas.

1º. ¿Cuál es la diferencia entre MODEM (modulador/demodulador) y un acoplador acústico, y qué hay que hacer para conectar uno de estos aparatos a un Commodore-64?

2º Sería posible conectar una guitarra al C-64? ¿Dónde y cómo?. Gracias.

F. C. T.
BARCELONA

El MODEM es el circuito que convierte los ceros y unos de tu ordenador en tonos de audio-frecuencia para que puedan ser transmitidos por la línea telefónica, grabados en un magnetófono normal o emitidos por una emisora de radio, mientras que el acoplador acústico es uno de los dispositivos que sirven para conectar el modem al teléfono en esencia no es más que un micrófono y un auricular, que colocados junto a los del teléfono permiten "acoplar" las señales "acústicas".

Lo de acoplar el C-64 a una guitarra es posible, sólo tendrías que saber exactamente lo que quieres hacer, diseñar un interface apropiado y el programa correspondiente. Las aplicaciones y posibilidades de un microordenador están limitadas por la habilidad e imaginación del programador y los periféricos que le conectes.

DE TODO UN POCO



Soy un lector de vuestra revista y tengo un VIC-20, y quiero haceros varias preguntas acerca de él:

1º. Cuánto puede costar un modem y si Commodore tiene uno propio.

2º. Cuáles son las instrucciones para controlar el "joystick"

3º. Tengo unos programas para el VIC que no los he hecho yo, sino que los he copiado de otras revistas, ¿puedo enviarlos para que los publiquéis? Algunos son muy buenos.

4º. Cuánto vale el cartucho "super-expander"

5º. Si al crear nuevos caracteres en el VIC ¿se pueden imprimir con la impresora?

6º. ¿Qué cintas aconsejáis para crear una biblioteca de programas?

No tengo más preguntas que haceros pero sí quiero daros esta sugerencia: después de comprobar un programa, podriais sacar la lista por la impresora y publicarla. Evitando así los errores de imprenta.

FRANCISCO JAVIER RECIO LLAMATA
FERNAN CABALLERO, 22
SEVILLA-1

1º No tenemos precios de modems en

NTONES.



España, si alguna casa los tiene esperamos que te los envíen por correo y a nosotros también.

2º Para leer la posición del Joystick debes utilizar los PEEK (37137) y PEEK (37152).

3º Aunque sean buenos es mejor que no los mandos, no podríamos publicarlos por no tener los "copyrights".

4º Creo que vale 9.600 ptas.

5º Efectivamente, el manual de la impresora te explica cómo hacerlo.

6º No podemos aconsejarte sobre la compra de cintas para tu biblioteca, sería perjudicar a unos anunciantes en beneficio de otros, y además el tipo de programas que deberías comprar depende de tus gustos y las aplicaciones a que quieras destinar tu ordenador.

Los programas salen listados directamente de impresora y son publicados así para evitar errores de imprenta, están comprobados todos, pero hay veces que una determinada secuencia de teclas, en un nivel, etc. puede ocasionar un error que no encontramos al probarlo, pero en cuanto nos damos cuenta lo publicamos en "meteduras de patas", somos humanos y tenemos fallos, intentaremos ser casi perfectos...

EDICION DE LINEAS



Me gustaría haceros algunas preguntas, en primer lugar me han hablado mucho en el sentido de que al escribir un programa en CBM-64, cuando dos líneas son muy parecidas, tiene una gran facilidad y yo todas las líneas las tengo que repetir una a una, no veo, dónde está la ventaja. También os pediría que escribiéseis algún programa para que al repasar un programa largo no fuese tan tedioso escribir línea por línea, sé que para el VIC-20 sí que lo hay. Me gustaría saber si ha salido el CURSO DE INTRODUCCION AL BASIC 2.ª PARTE, puesto que el primero te deja un poco desinformado, sobre todo a los profanos como yo.

BERNARDINO RODRIGUEZ SEGURA
BATERIA, 30-40, 3º D
MONCADA-BIFURCACION (Barcelona)

Programas que se escriban sólos hay poquísimos (tengo sólo uno: Crear un VIC-CASSO del número 4 de Commodore World), pero si te refieres a la instrucción AUTO, necesitas el cartucho correspondiente a tu ordenador: el de Ayuda al programador para el VIC-20 o su equivalente para tu C-64 (como el Simon's Basic de Microelectrónica, el Vic-Tree, etc.).

UTILIZACION DE CASSETTE NORMAL



Le estaría muy agradecido si con las revistas de COMMODORE WORLD, me enviara si fuera posible el esquema del interface para conectar un cassette grabador/reproductor de tipo normal.

Sin más por el momento se despide esperando sea atendida mi petición.

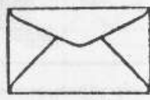
ENRIQUE LAIN SIMON
LEOPOLDO ROMEO, 19
ZARAGOZA-2

Existe un interface de ese tipo en el comercio, pero al igual que los aparecidos en algunas revistas, su funcionamiento no es del todo satisfactorio.

El problema reside en que un grabador de audio tiene en su interior circuitos, para protegerle contra la saturación (A.L.C.), y el Datassette hace todo lo contrario: grabar saturando siempre para evitar errores.

Si lo deseas podemos publicar un esquema de uno que yo utilicé durante mucho tiempo por falta de "cucas", pero me cansé de los "LOADING ERROR" y compré el cassette del VIC.

¿IRQ en "VIC-EXPANSION"?



En tres artículos publicados en la revista "Your Computer" (oct. nov. dic. 1983) aparecía desarrollada una ampliación del BASIC para el VIC-20, basada (si es que no entendí mal) en el vector IRQ. ¿Cómo se puede conseguir efectuar tal ampliación, por ese u otro método, en el CBM-64? Otra pregunta que me gustaría que me contestáseis es: ¿cuáles son las direcciones de las principales rutinas del intérprete BASIC en dicho aparato? Os agradecería que me contestáseis estas preguntas, o bien publicáseis artículos relacionados con estos temas, apenas tratados en la "Guía del usuario".

ANTONIO RIVAS MENENDEZ
CARTAGENA, 5-3º DCHA.
LA CORUÑA-6

Lo primero espero que lo tengas en tu poder por correo.

En inguno de los tres números que mencionas he visto que hable sobre modificar el vector IRQ. (Tengo los tres números y los listados vienen repetidos todos en el último, pero tienen errores,... pruébalos... y tranquilo, no te enfades con el autor ni con la revista... pasa en las mejores familias...).

Sobre el modo de ampliar el Basic de tu C-64 intentaremos publicar artículos en los que lo puedas ver. El sistema es el mismo que en el disk-O-Vic publicado en el número 2, sólo varían algunas direcciones de algunas rutinas.

Sobre las rutinas de cálculo del intérprete BASIC aparecieron artículos en los números 11 y 12 de Club Commodore, y nosotros iremos publicando lo que podamos.


Las Aventuras de RAMY ROMO
y sus amigos Kbit y Kbita
¿OYE RAFS QUE TAL SI NOS VAMOS TU Y YO A DAR UNA VUELTA?
¡PERFECTO! IREMOS A VISITAR ALGUN LUGAR INTERESANTE DENTRO DEL COMMODORE
¿OS PUEDO ACOMPAÑAR? ¡ASI PODEMOS IR TODOS EN BUS!
¿EN BUS?
COMMODORE WORLD


-SI, LA INFORMACION VA DE UN SITIO A OTRO DEL EQUIPO, GRACIAS A UNOS CANALES MUY RAPIDOS QUE SE LLAMAN BUSES
..HAY TRES.. UNO PARA LOS DATOS.. OTRO PARA LAS DIRECCIONES.. Y UN ULTIMO DE CONTROL. CADA UNO ENVIA UN TIPO DE INFORMACION ENTRE TODOS LOS RINCONES DEL EQUIPO


HALA VAMOS QUE SE NOS VA..
VOLANDO
ENTR.
COMM 2132-B
BUS DE DATOS
CHUIICCC


VAMOS EN UNO DE DATOS, YA QUE YO SOY UN DATO.. FIJAROS QUE RAPIDO VA... Y SIN ATASCOS


AHH
POM
POM
BORROMBLOM
CUCUUU
703
LAVE SUS DIENTES CON ESTRO
ÑIIIIIIIIIICC


PERO.. SI.. SSSIIII..SON...
¡PIRATAASS!
BUS


RAPIDO HUYAMOS.. ¡CIELOS, ESTAMOS PERDIDOS!. TODO EL ORDENADOR ESTA PERDIDO
¿PERO SE PUEDE SABER QUE PASA Y PARA QUE CORREMOS COMO TONTOS?
A ELLOOSS
SOS
S.O.S


SON PIRATAS.. VIENEN DEL EXTERIOR Y QUIEREN ROBAR EL MAPA DEL EQUIPO .. SI LO CONSIGUEN LO VENDERAN, PARA QUE OTROS FABRIQUEN EQUIPOS PARECIDOS Y TIMAR AL PUBLICO
¿?
POM
POM
POM


¡HUM! YA VEO.. SE DEDICAN A COPIAR MAQUINAS, PROGRAMAS Y TODO LO QUE VEN.. PARA LUEGO VENDERLOS BAJO OTRO NOMBRE
EXACTO
ESO.. ES.. SER UN PIRATA
¡VALE EINSTEIN!


PUES ENTONCES ESTA CLARO VAMOS A ENFRENTARNOS CON LOS PIRATAS
PERO ESO VA A SER MUY PELIGROSO
HUYY
¡CLARO!
ANDANDO
VENGA
MOVEROS


AAAHHHHH
¡UN LORO!


GROUAAAKKK ENEMIGOOOS GROUAAKKK
¡ES EL LORO DE LOS PIRATAS! VA A AVISARLOS DE QUE ESTAMOS AQUI... ¡DETENEDLOOOO!
VOY
AHH
ZIUUUUUUU


VERAS QUE PRONTO LO PARO EN SECO


POF
POM
GOUAGGRKK
ZIUUUU
ANDA QUE TIRO MAS TONTO


VAMOS RAFS DESPIERTA... NO HA SIDO MAS QUE UNA CHINITA ... VAMOOOS
¡AY! ¡CIELOS! ¡HORROR!
GLLLL
GRSSS
GROUUAK


LO LLEVAREMOS ENTRE LOS DOS HASTA QUE SE RECUPERE ... VAMOS A POR LOS PIRATAS


DOS DEDITOOOS JON...LOS QUE MEMIRAAAN
ALLI HAY UNO MONTANDO GUARDIA SI PUDIERAMOS ATRAPARLE LE INTERROGARIAMOS
FSSSS GLLLL


HOLAAA MUCHACHO COMO VA TU CUERPO... JI, JI, JI.. QUE OJOS MAS NEGROS TIENES
¡POR TODOS LOS BATRACIOS! ¿QUE PASA AQUI?
RAFS QUIETO NOO
GROOUK


NO ME GUSTA... GSSFFHH TU CARA FORASTERO, TE CONVERTIRÉ EN UNA FLOR... HIPPSSS ¡HOP!
ZKPOUS


CIELOS AHORA NO PODREMOS INTERROGARLE
PERO QUE ME HA PASADO.. EN MIS DEDITOS... GSSSS.GÑN.. QUE FLOR MAS RARA ME HA SALIDO..
¿?


¿?
¡!
SE ESCAPA LA CABRA.. SIGAMOSLA IRA DONDE LOS OTROS PIRATAS
VAMOS
BEEE
ZIUUU


CIELOS.. SE HA DADO OTRO GOLPE
CROC
GRAAZIXUIN


PERO... ¿QUE HA PASADO? ¿DONDE ESTÁ EL LORO?
ANDÁ YA SE HA RECUPERADO
HUYYY
AY

PUES ENTONCES VAMOS TRAS EL LORO, PARA QUE NOS LLEVE DONDE LOS PIRATAS

POR TODOS LOS BATRACIOS DEL CARIBE
CAMPAMENTO PIRATA
ZIUUU

¡BIEN MUCHACHOS.. ESTAMOS DE SUERTE.. AQUI ESTAN LOS PLANOS! ¡LOS VENDEREMOS Y NOS VAMOS A FORRAR A TOPE!
COMMODORE
PLANOS

VIVA NUESTRO JEFE...
YUPPIII
VIVAAA
OLE
NOS VAMOS A LLENAR DE ORO
EL MEJOR CORSARIOO
CHOMP CHOMP

HURRAAAAAA
¿?
OLEE
BRAVO
YUPPI
CHOMP CHOMP

BIENNN
BRAVOO
CHOMP

LOS PLANOOOS SE LOS HA COMIDO UNA CABRAAA
CHOMP

ANDÁ LA CABRA OTRA VEZ
¿QUE CABRA?
VIENEN LOS PIRATAAASS
BEEEE

GROUUAAAK
LOS PIRATAAASSS
A ELLOS MIS VALIENTEESS
BEEEEE

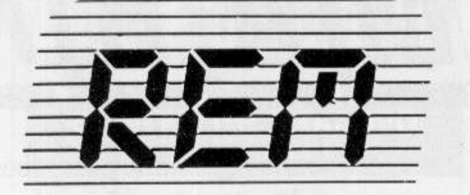
CUCUUU... HOLA PIRATAS

CIELOS EL EMPERADOR HUYAMOSSS
LOS SOLDADOS HUYAMOOS
¿SE PUEDE SABER POR QUE VIENE LA CABRA CON NOSOTROS?
BEEE

EHH... VOSOTROS ESPERAD UN MOMENTO ... ¡HE DE HABLAROS UN INSTANTE...! ¡VOLVEDDD!
Y UN JAMON... EMPE.. Y.. UN JAMON
ZIUUUUUU
ERRAZQUIN

Y SALEN CORRIENDO.. ¿?... AHORA QUE QUERIA FELICITARLOS POR PELEAR CONTRA LOS PIRATAS.. ¡ESTAN TODOS LOCOS!

VEN A LA TIENDA Nº1 DE MADRID
REMSHOP-1
HORARIO: 10 MAÑANA A 10 NOCHE DE LUNES A SABADO



REM Somos profesionales.
REM Nace para dar mejor servicio.
REM Como oferta de lanzamiento, entrega un Bono del 10% del importe de su compra para adquirir nuevo material o programas.
REM CAMBIO acepta equipos de 2.ª mano al adquirir otro nuevo.
REM Consúltanos tus necesidades.

RENOVACION EN MARCHA, S. A.
c/. Espronceda, 34-2º int. - MADRID-3
Teléfono **(91) 441 24 78**

REM SHOP 1
c/. Galileo, 4 - MADRID-15
Teléfono **(91) 445 28 08**

HARD COMMODORE 64 JUEGOS

1 Commodore 64	79.900 Ptas.
1 Cassette	12.000 Ptas.
1 Joystick (puño)	4.000 Ptas.
2 Paddle	2.500 Ptas.
TOTAL	98.400 Ptas.

PRECIO PTAS. TOTAL 88.560

HARD COMMODORE 64 UTILID.

1 Commodore 64	79.900 Ptas.
1 Cassette	12.000 Ptas.
1 Impresora GP 500 + Interface	69.900 Ptas.
1 Unidad de Disco (170 K)	95.000 Ptas.
TOTAL	256.800 Ptas.



SOFT COMMODORE 64 JUEGOS TOP TEN

Programa	Precio
FUTBOL (cartucho)	6.000 Ptas.
LE MANS (cartucho)	4.500 Ptas.
DRAGONSDEN (cartucho) joystick)	4.500 Ptas.
BUGABOO (cinta)	1.900 Ptas.
CUDDLY CUBURT (cinta)	1.900 Ptas.
REVENGE MUTANT CAMELS (cinta)	2.200 Ptas.
KONG (cinta)	1.800 Ptas.
PANIC 64 (cinta)	1.900 Ptas.
HUNGRY HORACE (cinta)	2.200 Ptas.
DEFENDER 64 (cinta)	1.700 Ptas.

PRECIO TOTAL 26.190 PTAS
INCLUYENDO COMMODORE 64 + CASSETTE + JOYSTICK + PADDLE
114.750 PTAS

SOFT COMMODORE 64 UTILIDADES/EDUCACION

Programa	Precio
CONTABILIDAD PERSONAL (cartucho + disco)	10.000 Ptas.
EASY SCRIPT (cartucho + disco)	20.000 Ptas.
CALC RESULT (cartucho + disco)	33.000 Ptas.
PROGRAMMER'S UTILITIES (disco)	7.500 Ptas.
BASE DATOS (cinta)	3.300 Ptas.
MAILING Y ETIQUETAS (cinta)	3.300 Ptas.
ULTRABASIC (cinta)	2.900 Ptas.
MUSIC COMPOSER (cartucho)	4.500 Ptas.
VISIBLE SOLAR SYSTEM (cartucho)	4.500 Ptas.
CP/M (cartucho + disco)	20.000 Ptas.

PRECIO TOTAL 98.100 PTAS
INCLUYENDO COMMODORE 64 + DISCO
255.510 PTAS

REM NOTICIAS

REM FRANCHISING
Si quieres montar tu propia minitienda de informática, o una tienda especializada, envíanos tu dirección y recibirás información completa.

REM DETALL
Si quieres vender nuestros productos, envíanos tu dirección y recibirás puntual información.

REM SOFT
Pagamos 20% de royalties de programas originales.

REM COMMODORE CLUB
Reuniones. Intercambio de programas. Revistas. Libros. Unete a nosotros.

REM INFORMATICA
Tenemos también SPECTRUM, SPECTRAVIDEO, ATMOS, ATARI y MULTITECH Hard y Soft.

BOLETIN DE PEDIDO

Nombre y apellidos ....
Dirección y Teléfono ....
Deseo recibir más información ....
Deseo adquirir ....
Precio total (incluye 300 Ptas. de gastos de envío) ....
Giro Postal ☐ Giro Telegráfico Transferencia Bancaria
Ingreso en cuenta 3769 8 BANCO DE BILBAO. Ríos Rosas, 44. MADRID-3
Talón adjunto Talón conformado adjunto
Tarjeta VISA número ....
Fecha Caducidad .... Firma

REM c/. Espronceda, 34-2º int. - MADRID-3

MARKETCLUB

Servicio gratuito para nuestros lectores particulares. Empresas 300 ptas., por línea.
De no especificar duración, los anuncios serán publicados durante 1 mes.

MERCADILLO

• Compro programa en cassette para VIC-20 de cuenta de explotación con análisis de gastos. Tel.: 93/337 20 20. Contactar con Angel Amores C/ Sant Roc, 12, 2º, 9º. **HOSPITALET DE LLOBREGAT** (Barcelona). ***(Ref. M-47)***

• Intercambiamos programas de todo tipo para el VIC-20. Llamar de 20 a 22 horas, preguntando por Aitor o Ignacio. Tel.: (943) 54 11 02. C/ Emeterio Arrese, 5, 4º B. **TOLOSA** (Guipúzcoa). ***(Ref. M-48).***

• Vendo VIC-20 comprado en marzo-83, cartucho de ampliación 3 K, las dos partes del curso de basic, 1 cartucho de juegos, varios juegos en cassette, todo en perfecto estado, sólo por 30.000 ptas. Luis Llenas. Ctra. Palamós, 34. **PLAYA DE ARO** (Gerona). Tel.: (972) 81 76 51. ***(Ref. M-49).***

• Vendo Commodore 64, unidad de disco VIC 1541, MASTER 64, disco 6 juegos, disco música 64, caja 10 diskettes vírgenes Commodore, 2 joystick, guía de referencia del Commodore 64 y libro con más de 50 programas listados. Comprado enero-abril 84. Todo por 175.000 ptas. José Escudier. C/ Utrera, 6. **CADIZ**. Tel.: (956) 25 27 67 (tardes). ***(Ref. M-50).***

• Vendo CBM64 - cassette - manual referencia programador joystick - revistas club Commodore - 4 juegos (Kick off, Mutans Camels, Frogger, Jam breaker) con 3 meses, por 85.000 ptas.; precio en el mercado 107.400 ptas. Antonio Cebrián Martínez. C/ Francisco Pizarro, 73, 3º D. **ALBACETE.** ***(Ref. M-51).***

• Vendo/compro/intercambio programas para el VIC-20 no ampliado (cintas y fotocopias). Francisco Javier Recio Lamata. C/ Fernán Caballero, 22. **SEVILLA-1.** ***(Ref. M-52).***

• Intercambio programas de todo tipo para un VIC-20 ampliado o sin ampliar. Escribir a Daniel Bassas. **GERONA** capital. Plza. Marqués de Camps, 16, 3º, 2.ª. Tel.: 21 27 26. ***(Ref. M-53).***

• Vendo VIC-20, con cassette Commodore, expansión de memoria de 16 K, super expander para gráficos de alta resolución, curso de introducción al Basic (Parte I), programas, juegos. Comprado en octubre del 83, buen estado. Precio: 55.000 ptas. Antonio Utrera Conejo. C/ Cister, 6. **MALAGA**. Tel.: 21 51 20. ***(Ref. M-54).***

• Usuario microordenador VIC-20 desearía contactar con colegas para realizar intercambio de programas de todo tipo y de capacidad RAM. Rafael O'Donnell Verger. C/ 31 de Diciembre, 43, 1º, 2.ª. **PALMA DE MALLORCA-3**. (Baleares). ***(Ref. M-55).***

• Vendo VIC-20, 1 cartridge, guía de referencia del VIC, algunos programas. Comprado 9 de septiembre 83. Precio 30.000 ptas. Javier Boneu Bonet. Avda. Blondel, 33. **LERIDA**. Tel.: (973) 27 01 93. ***(Ref. M-56).***

• Vendo programador universal de memorias EPROM, módulos 32K RAM dinámicas, placas ampliación ROM y diversos accesorios y juegos para VIC-20. Compraría unidad de discos. Félix Portabella Padro. C/Forn de Santa Lucía, 1-2º. **MANRESA** (Barcelona). Tel.: (93) 872 22 97. ***(Ref. M-57).***

• Vendo VIC-20 (en buen estado), cassette, ajedrez, introducción al Basic I y los manuales por 42.500 ptas. Jorge Llobet Ribeiro. C/ Guillermo Tell, 38 pral. 2.ª. **BARCELONA-6**. Tel.: (93) 200 66 32. ***(Ref. M-58).***

• Esta es mi primera carta que les mando después de la de la suscripción. Su revista es francamente buena; yo aún no he colaborado por falta de tiempo y sobre todo por carecer de: "Datassette". Tengo 16 años, muy pocos meses con mi VIC-20 y mi presupuesto no me da para más, por lo cual deseo publiquen el siguiente llamamiento a ver si hay suerte, en la sección de Mercadillo.
A todo amable lector que su antiguo "Datassette" se le haya roto o agrietado, o tenga uno que le va mal o simplemente de sobra, estaría muy agradecido de recibirlo. Como aliciente le "cambiaría" o regalaría un interesante interface, catálogos de hard-software, etc. Gracias. Máximo Martín. Apartado 1185. **LA CORUÑA.** ***(Ref. M-59).***

• Vendo consola de videojuegos CBS Colecovisión con tres cartuchos. La mejor del mercado. Como nueva (diciembre 1983). Sólo por 25.000 ptas. Interesados de Barcelona llamar horas fuera de oficina, al Teléf.: 301 49 37. **BARCELONA**. C/Santa Ana, 10, 2º 1.ª. Manuel Pons Pons. ***(Ref. M-60).***

• Vendo VIC-20 comprado en mayo 83 totalmente nuevo + unidad de cassette (Dataset) + cartucho Super Expander + las dos partes del curso Basic + las cintas + cintas con varios programas de juegos educativos y de gráficos. Los programas también podría mandar con los listados. Todo por 45.000 ptas. Raimundo J. Lorenzo Peña. C/ Ríos Rosas, 3, 2º D. Tel.: 22 06 18. **ALBACETE.** ***(Ref. M-61).***

• C-64, intercambio programas, experiencias, etc. Javier Salas Varela. C/ Velázquez, 1. **CADIZ-10.** ***(Ref. M-62).***

• Vendo Commodore Vic-20, comprado en enero del 84. Garantía vigente. Está nuevo. Incluye cassette commodore, 2 cintas de programas, la suscripción a esta revista y otras 20 revistas más. Llamar al (94) 416 49 04. Preguntar por Asier. ***(Ref. M-63).***

• Vendo VIC-20 (comprado en enero-84), con cassette, manual del usuario y diversos juegos y programas de aplicaciones por 40.000 ptas. Llamar a partir de las 17 horas. Salvador Farres Magem. C/ Montaña, 55 bjos. **VILADECANS** (Barcelona). Tel.: (93) 658 02 60. ***(Ref. M-64).***

• Vendo o cambio programas para Commodore 64. Gráficos, assembler, proceso de textos, juegos, etc. Luis Sánchez Pérez. Apartado 176. Tel.: 22 14 94. **TOLEDO. (Ref. M-65).**

• VIC-20 nuevo, vendo por haberme pasado a ordenador de mayor potencia.
MICROORDENADOR VIC-20 (enero 1984) por 25.000 ptas. Cassette nuevo por 10.000 ptas. Basic 1 y 2 (libro y cintas juntos) 5.000 ptas. Varios complementos (libros VIC-20, juegos, cintas ficheros, datos y contabilidad del hogar en castellano y en inglés, etc.) por 10.000 ptas. Ampliación de 16 K nueva por 10.000 ptas. Se vente todo junto por 60.000 ptas. En unidades sueltas, precio a convenir. Llamen Srta. Marina. Tel.: 380 35 44. **BARCELONA.** ***(Ref. M-66).***

• Quisiera comprar y cambiar el cartucho super expander por 4.000 ptas. junto con un cartucho de juego, o por muchos cuentos de super héroes y monopatín, o por el scalextric. GP-27 ampliado. Agustín Ruiz Castrillo. C/ Gregorio Solabarrieta, 49, 4º Izda. **MIRANDA DE EBRO** (Burgos). ***(Ref. M-67).***

• Usuario de C-64, estoy interesado en el intercambio de experiencias y programas de todo tipo. Viviendas Santiago Apostol, **125. 1º Dcha. NARON** (El Ferrol). Teléfono: 38 58 71. ***(Ref. M-68).***

Vendo un COMMODORE 64, nuevo a estrenar, con garantía, con Cassette Commodore, con cartucho de fútbol, con cinta de ajedrez "Gran Master" y un libro con 60 programas para el CBM-64, y todo por 65.000 ptas. Juan Benítez Sánchez. Pº Fabra y Puig, 331. Tel.: 358 98 06. **BARCELONA-31.** ***(Ref. M-69).***

CLUBS

• Club de amigos del VIC-20 y Commodore 64. Ramón O. Serna Soler. Foto Estudio 2. Plaza de Sombrereros, 2. **PALMA DE MALLORCA.** Tel.: 21 31 62. ***(Ref. C-1).***

• Desea información sobre clubs en existencia y gente que quiera formar uno. Agustín Quevedo Velasco. C/ Oña, 55-4º B. **MADRID-34.** Tel.: 202 94 28. ***(Ref. C-2).***

• Desea información sobre clubs en existencia y gente que quiera formar uno. Pablo Nistal Alonso. C/ Empecinado, 46, 6º B. **MOSTOLES.** Tel.: 645 21 70. ***(Ref. C-3).***

• Club de programación Alaiz, en Pamplona. Está dirigido a estudiantes de B.U.P. y C.O.U. de esta ciudad, y el fin que tiene es la introducción en el mundo de la informática y programación BASIC. No tiene, por lo demás, ningún fin lucrativo. El club propiamente posee equipos COMMODORE 64. VIC-20 y diversos periféricos (impresora, unidad de disco y cassette). CLUB ALAIZ. Pza. Monasterio Santa Gemma, s/n. Tels.: 254 480-257 704. **PAMPLONA.** ***(Ref. C-4).***

• Club de Commodore de **Albacete**. Fernando Martínez Guerrero. La Roda, 39. 5º D. **ALBACETE.** ***(Ref. C-5).***

• Círculo de Durensan Vigués. **VIGO**. C/ Venezuela, 48. Entrechan. Tels.: (986) 410 683/422 519.. ***(Ref. C-6).***

• Un grupo de usuarios de la Comunidad Autónoma de Murcia intenta crear un Club de Usuarios de VIC-20 y C-64 en esta región, sin fines lucrativos. Interesados llamar a los teléfonos (968) 80 23 64 (Mañanas, preguntar por Paco); (968) 26 72 62. (Tardes, preguntar por Fernando). Francisco Sáez Soto. C/ Virgen del Rosario, 28. **ALCANTARILLA**. Murcia. ***(Ref. C-7).***

• Deseo contactar con interesados en formar un club de microordenadores en Barcelona e intercambiar experiencia. Domenec Garrofe. C/ Aragón, 575, 4º 2º. **BARCELONA-26**. Tel.: 348 00 39. ***(Ref. C-8).***

• Interesados en la creación de Club de Usuarios en Girona escribir a "Apartado de Correos nº 580" de **GIRONA.** ***(Ref. C-9).***

• Xabier Prat i Burdo que vive en c/ Rabal Sta. Eulalia s/n. de Sta. María Délo **(BAGES)** Barcelona, quisiera saber si existe algún Club Commodore cerca de Manresa y solicta información sobre todo lo relacionado con Modems. ***(Ref. C-10).***

• Os rogaría me comunicárais si existe algún Club Commodore en mi ciudad **(GUADALAJARA)** o en **ALCALA DE HENARES** (donde trabajo). También agradecería información acerca accesorios, publicaciones, etc., así como de distribuidores de Madrid que trabajen con el C-64. Miguel Angel Capeña Alonso. C/ Tabernero, 16-3º I. Tel.: (91) 22 67 57. ***(Ref. C-11)***

• Desearía información sobre la existencia de algún Club Commodore en Somaén o sus cercanías. Carlos Cabrera. Camino de las Eras, s/n. **SOMAEN** (Soria). ***(Ref. C-12).***

• Asociación Manchega de Usuarios de CBM-64 (A.M.U.C.-64). C/ Blasco de Garay, 10. **ALBACETE**. Esta asociación persigue extender el uso racional de la informática, concretándose de momento en el ordenador Commodore CBM-64. ***(Ref. C-13).***

• Os agradecería que comuniquéis a los lectores de la revista el nacimiento del Club MICRO-ESPLAI, en el que nos reunimos aficionados y profesionales de la electrónica e informática con ganas de divertirnos aprendiendo e intercambiar experiencias.

Estamos en Gran Vía, 682, Pral. (Entre Bruch y Gerona) de **BARCELONA**. Todos los amigos que tengáis más de 15 años y menos de 100 podéis participar en esta experiencia. Y si sois más jóvenes, venid también, ya nos las ingeniaremos para organizar actividades interesantes.

Para informaros, los viernes de 6 a 8 tarde o los sábados de 9 a 11 de la mañana. ***(Ref. C-14).***

• Queremos formar un Club Commodore en Barcelona. José. Tel.: 329 86 31. C/ Aurora, 19-4º 1.ª. **BARCELONA.** ***(Ref. C-15).***

• Me dirijo a ustedes por un hecho reciente, que es la disolución del único club de usuarios en Barcelona con sede en ELEKTROCOMPUTER. Quiero decirles que su revista es un medio para crear un nuevo Club, por eso pido que si ustedes lo creen conveniente, publiquen esta carta. Eduardo Macip Fite. C/ Horno de la Villa, nº 4. **MONTBLANC.** Tarragona. ***(Ref. C-16).***

• Quisiera saber si existe algún club commodore en Sevilla. Francisco Javier Recio Lamata. C/ Fernán Caballero, 22. **SEVILLA-1.** ***(Ref. C-17).***

• Agradecería mucho a todo el que pudiera informarme sobre la creación del club commodore en Girona. Daniel Bassas. **GERONA** capital. Plaza. Marqués de Camps, 16, 3º, 2.ª. Tel.: 21 27 26. ***(Ref. C-18).***

• El club Ensenada de Santander ha puesto en marcha un club de usuarios Commodore, disponemos de ordenadores Commodore 64 y VIC-20 dotados de diversos periféricos. Su actividad se centra en el intercambio de programas y experiencias, perfeccionamiento de lenguajes, uso del logo, etc. Club Ensenada. C/ Sta. Lucía, 1, 1º. Tel.: 22 59 92. ***(Ref. C-19).***

• Estoy formando un club de intercambio de programas (en cinta) para VIC-20. Jaime Vidal Forteza. C/ Sor Damiana, 16-1º. S. Francisco (Pil. lari). **PALMA DE MALLORCA.** ***(Ref. C-20).***

• Me interesaría conocer si hay algún club de amigos del VIC-20 cerca de mi ciudad. Agustín Ruiz Castrillo. C/ Gregorio Solabarrieta, 49. 4º izda. **MIRANDA DE EBRO** (Burgos). ***(Ref. C-21).***

BOLSA DE TRABAJO

OFERTA DE TRABAJO ¡URGENTE!

SE NECESITA VENDEDOR DE PUBLICIDAD, DINAMICO CON EXPERIENCIA (RESIDENTE EN MADRID)

Contactar con:
COMMODORE WORLD
C/Pedro Muguruza, 4. 8º B
Madrid-16. Teléf.: 259 54 78

OFERTA DE TRABAJO ¡URGENTE!

PROGRAMADOR (...O CASI...)

DIEGO NECESITA AYUDANTE (QUE SEA BUENO/A) PARA COMPROBACION DE PROGRAMAS, QUE QUIERA APRENDER MUCHO Y TRABAJAR UN MONTON, RESIDENTE EN MADRID Y BUENA COMPRENSION DEL INGLES (...NO NECESITA HABLARLO...)

Contactar con:
COMMODORE WORLD
C/Pedro Muguruza, 4. 8º B
Madrid-16. Teléf.: 259 54 78

DEX

Decrementa el registro X en una unidad

Operación: X—1 → X

(Ref.: 7.6)

N Z C I D V
√ √ – – – –

Modo de Direc.	Formato en ensamblador	Código Operan.	Núm. Bytes	Núm. Ciclos
Implícito	DEX	CA	1	2

DEY

Decrementa el registro y en una unidad

Operación: Y—1 → Y

(Ref.: 7.7)

N Z C I D V
√ √ – – – –

Modo de Direc.	Formato en ensamblador	Código Operan.	Núm. Bytes	Núm. Ciclos
Implícito	DEY	88	1	2

EOR

"Or exclusivo" de la memoria con el acumulador

Operación: A ∀ M → A

(Ref.: 2.2.3.2)

N Z C I D V
√ √ – – – –

Modo de Direc.	Formato en ensamblador	Código Operan.	Núm. Bytes	Núm. Ciclos
Inmediato	EOR #Oper.	49	2	2
Pág. Cero	EOR Oper.	45	2	3
Pág. Cero, X	EOR Oper., X	55	2	4
Absoluto	EOR Oper.	4D	3	4
Absoluto, X	EOR Oper., X	5D	3	4*
Absoluto, Y	EOR Oper., Y	59	3	4*
(Indir., X)	EOR (Oper., X)	41	2	6
(Indir.), Y	EOR (Oper). Y	51	2	5*

* **Suma 1 si se salta de página.**

INC

Incrementa la memoria en una unidad

Operación: M + 1 → M

(Ref.: 10.6)

N Z C I D V
√ √ – – – –

Modo de Direc.	Formato en ensamblador	Código Operan.	Núm. Bytes	Núm. Ciclos
Pág. Cero	INC Oper.	E6	2	5
Pág. Cero	INC Oper., X	F6	2	6
Absoluto	INC Oper.	EE	3	6
Absoluto, X	INC Oper., X	FE	3	7

El Misterio de los Pokes (IV)

Por Diego ROMERO



Como prometimos en el número anterior, vamos a tratar de aclarar la función de esos "misteriosos pokes" que aparecen en muchos programas al utilizar caracteres definidos por el usuario, me refiero a los punteros de principio y final de memoria ocupada por el programa en BASIC, sus posiciones de memoria y funciones son los de la tabla 1:

Dirección Hex	Dirección Dec	Nombre	Función
2B-2C	43-44	TXTTAB	Puntero de inicio BASIC.
2D-2E	45-46	VARTAB	Puntero de inicio de variables.
2F-30	47-48	ARYTAB	Puntero de inicio de tablas.
31-32	49-50	STREND	Puntero de fin de tablas +1.
33-34	51-52	FRETOP	Puntero de inicio de cadenas.
35-36	53-54	FRESPC	Puntero de utilidad para cadenas.
37-38	55-56	MEMSIZ	Puntero de final de RAM.

Como podéis ver todas estas direcciones van asociadas dos a dos, es así por tratarse de vectores, y como ya sabréis para poder señalar a cualquiera de las 65536 posiciones de memoria de nuestro ordenador necesitamos dos bytes. El modo en que se almacenan los vectores es el que ya expliqué en el cursillo de lenguaje máquina del número anterior. Para aquellos que no lo sigan repetiré que se almacena en la primera posición de memoria el byte de menor peso del vector, y en la siguiente posición de memoria al byte de mayor peso.

En la tabla aparecen unos nombres "raros" que son los nombres que les dio la casa Commodore al diseñar el ordenador, es mejor no fijarse mucho en ellos, de momento sólo nos interesan las funciones de cada vector.

El primer vector (TXTTAB) es el que apunta al principio de la memoria de reservada para el programa BASIC, en el VIC 20 varía según la ampliación de memoria que tengamos, y en el C-64 suele apuntar a la dirección $0801, por lo que la dirección 43 contiene un uno y la 44 contiene un 8. Este vector se puede alterar cuando se utilizan gráficos de alta resolución para reservar espacio antes del programa BASIC.

El segundo vector (VARTAB) es el de final de programa BASIC y va subiendo según aumentemos el tamaño del programa. El final del programa coincide con el inicio de la zona reservada a variables, por eso se le denomina "inicio de variables". El vector va apuntando a posiciones más altas de memoria conforme vamos definiendo más variables en el programa.

El tercero (ARYTAB) es el del final de variables y coincide con el principio de las tablas de matrices. A partir de la posición de memoria a donde apunte, se guardan unas tablas de vectores, que apuntan a cada uno de los elementos de cada matriz que tengamos definida.

El cuarto vector (STREND) es el puntero que indica cuál es el final de la zona reservada a tablas de matrices. Incrementa su valor conforme vamos definiendo más matrices en el programa con sentencias DIM.

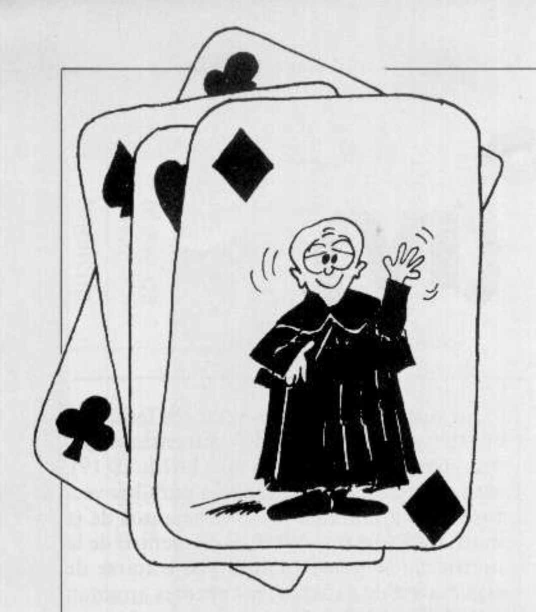
La distancia entre los vectores tercero y cuarto aumenta cuando dimensionamos una matriz, por ejemplo con DIM A$(19), esta sentencia reserva espacio para los vectores que apuntan a los 20 elementos de la matriz A$ (del 0 al 19). Los elementos de la matriz no se guardan aquí por tratarse de una matriz de cadenas, los vectoes apuntan a la dirección donde se encuentran los elementos de la matriz dentro del espacio reservado para cadenas.

El quinto vector (FRETOP) es el que señala el principio de la zona reservada para cadenas, en esta zona se realizan todas las operaciones que tienen que ver con los strings, por ejemplo: cuando decimos A$=A$+B$, la CPU busca la cadena B$, la A$ y las concatena dejando el resultado en el espacio reservado para cadenas. Este vector al revés que los anteriores, disminuye cuando se van introduciendo más cadenas, es decir, crece hacia abajo.

El sexto vector (FRESPC) es un puntero que utiliza la CPU para todo tipo de operaciones con cadenas, pero no señala a ninguna posición de memoria en particular.

Por último el séptimo vector (MEMSIZ) es el que señala al final de la zona utilizable para programas en BASIC. Si hacéis un POKE a estas posiciones y después borráis las variables con el comando Basic CLR para restaurar los demás punteros, podréis alterar la memoria libre para programas en Basic, probad diferentes valores en estas dos últimas posiciones, luego CLR y hacer PRINT FREE (1) con RETURN, ya veréis cómo podéis aumentar o disminuir la memoria libre.

Cuando se definen caracteres propios, se suelen modificar estos valores para proteger una zona de la memoria de la intrusión por el intérprete Basic. Si queréis obtener más información sobre el modo de hacerlo, podéis leer el artículo aparecido en la sección VIDEO-CASINO del número anterior. ■



VIDEOCASINO

Vive toda la emoción en una "guerra artística" Para la adquisición de unos cuadros de valor con este programa que presenta una subasta de arte. Podrías hacerte con un Rembrandt genuino.

SUBASTA

Las simulaciones, que se prestan a los juegos de ordenadores, proporcionan un amplio fondo para presentar juegos de lógica, las matemáticas y de otros tipos. Se puede construir cualquier tipo de concurso de habilidad o de pura suerte alrededor de una simulación.

Una simulación normalmente es un modelo computerizado de una situación y el jugador o jugadores desempeñando el papel de uno o más participantes en la situación que se esté simulando. Un escenario típico puede ser la operación de un puesto de venta, un viaje al oeste en carro, o, como en este juego, la compra y venta de una obra de arte en una subasta.

En todos los juegos de este tipo, tienes que tomar una serie de decisiones a lo largo del juego que tengan una influencia sobre el resultado. A menudo, se juega mejor cuando se dispone de unos conocimientos sobre la situación real.

Por ejemplo, se ha construido una situación compleja alrededor de un escenario tipo "Hammurabi", en que tú eres el jefe absoluto de un país y tienes que decidir las tierras que serán plantadas, los alimentos que serán dados a la población y la cosecha. De acuerdo con la sabiduría de tus decisiones, las consecuencias pueden ser la riqueza o la bancarrota. Además de las acciones del jugador, se utilizan unos factores aleatorios para determinar la cosecha, nacimientos, muertes, etc.

VIC-20
NO AMPLIADO
C-64

Los juegos de tipo aventura también son un tipo de simulaciones. Estos intentan imitar un viaje a través de un castillo embrujado, una central nuclear u otro sitio interesante. La mayoría de ellos tienen muy pocos, por no decir ninguno, factor aleatorio; tu meta es descubrir una serie de sucesos establecidos que conducen a la solución de la aventura.

"Obras Maestras" es un ejemplo del tercer tipo de simulación, en la cual la suerte controla por completo el resultado. Tú tienes que utilizar tu intuición, percepción o pura suerte para adivinar la combinación de precios y apuestas que proporcionarán los resultados más favorables.

El juego de enfrentar a unos contrincantes en una guerra para comprar y vender cuadros al mejor precio posible. La finalidad del juego es, al concluirse este, tener tú la mayor parte posible, o más de la cantidad original de $5000.

Te ofrecen cinco cuadros, uno por uno, que tú puedes comprar con tus $5000. Tú puedes pujar una vez para cada cuadro. El ordenador selecciona una cantidad al azar para pujar, que puede ser más alta o más baja que la tuya. Si tú has ofrecido más dinero, te quedas con el cuadro. Si no, lo pierdes para siempre.

Obviamente, una buena estrategia a seguir es asegurar que tú ganes ofreciendo la cantidad más pequeña posible, dejando así la máxima cantidad para las compras posteriores. Sin embargo, dado que es difícil pronosticar lo que el ordenador ofrecerá, esto puede resultar bastante complicado.

Después de que se hayan vendido todos los cuadros, tú puedes vender, los que hayas podido comprar. A medida que cada cuadro sale a subasta, la cantidad original que se pagó por él será indicado, y el ordenador le hará una oferta que puedes aceptar o rechazar.

Una cantidad superior puede o no ser ofrecida. El número de ofertas será aleatorio. Tú nunca sabrás si debes aceptar la oferta o esperar por si te ofrece un precio más alto. La oferta más reciente podría ser la última. En ese caso, te habrás quedado con el cuadro y no recibirás dinero al final de juego.

La parte del programa del C-64 que incluye la compra empieza en la línea 190, donde un bucle For... Next se repite de 1 a 5. Se representa el dinero restante (Cash), junto con el número del cuadro en venta y una invitación a pujar. No puedes ofrecer más dinero que el que tienes (línea 270).

A continuación, el ordenador hace la contraoferta (OP). El ordenador elegirá un número de 1 a 99, lo multiplicará por 10 y le sumará 150. De esta forma, la oferta más baja será de $160, y la más alta $1140. Es mejor si no sabes esto. De no ser así, ninguna oferta será mayor de $11401. Ten en cuenta que dado que se ofrecen cinco cuadros, no será posible ofrecer $1141 por cada uno, aún sabiendo tú el secreto.

El número de cuadros comprados, NU, se utiliza para limitar otro bucle For...Next, de 1 a NU, durante el cual se pueden vender los cuadros. El precio de compra de cada cuadro de tu colección ha sido previamente almacenado en un "array", Collect (NU) y será representado.

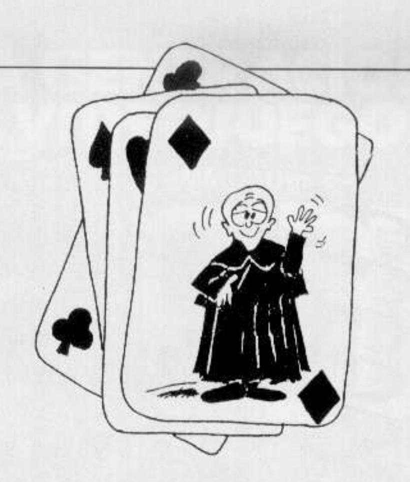
Un bucle anidado va desde el 1 al número de ofertas. Esta cifra, NB, se elige en la línea 460, y será de 1 a 7 ofertas. En algunas tandas, sólo se hará una oferta, en otras, se pueden hacer hasta siete. De esta forma, tú nunca sabrás si la oferta más reciente será la última, o si se hará una oferta más sustanciosa.

La cantidad ofrecida se determina en la línea 48Ø y puede ser de $500 a $2409. Aquí de nuevo, los jugadores no deben de saber estas cifras. Lo ideal es que nadie sepa cuál puede ser la oferta más baja ni la más alta. Si estos valores pueden ser deducidos después de haber jugado muchas veces, pueden ser cambiados haciendo unas modificaciones en el programa.

Por ejemplo, puedes modificar la cantidad mínima y máxima que tu contrincante puede ofrecer inicialmente por un cuadro determinado, o la cantidad que será ofrecida cuando el cuadro se pone en venta. También puedes aumentar el número de ofertas posibles de siete a diez, o cualquier otro número.

Si se acepta la oferta, la cantidad se suma a tu "Cash". Si se rechaza la última oferta, un mensaje aparece en pantalla. Una vez vendidos todos los cuadros, se presentan los resultados, y puedes volver a jugar.

El programa "Obra Maestra" proporciona toda la diversión de una subasta de verdad, pero no te cuesta dinero. Como simulación, constituye una introducción sencilla a un tipo de juego cada vez más popular. ■



Listado para el VIC-20 y el C-64.

```
10 REM******************
20 REM*                *
30 REM* OBRAS MAESTRAS *
40 REM*                *
50 REM******************
60 CASH=5000
65 POKE53281,1
70 PRINT"[CLR][2CRSRD]"
80 PRINTTAB(4)"[RVSON][RED]OBRAS[SPC]
MAESTRAS[BLU][RVSOFF][2CRSRD]"
90 PRINTTAB(2)"TU[SPC]DEBES[SPC]PUJA
R[SPC]POR"
100 PRINTTAB(4)"CINCO[SPC]CUADROS"
110 PRINTTAB(1)"VALIOSOS.[2SPC]COMEN
ZANDO"
120 PRINTTAB(6)"EN[SPC]5000[SPC]$."
130 PRINTTAB(1)"ENTONCES[SPC]DEBES[SPC]
VENDER"
140 PRINTTAB(1)"TU[SPC]COLECCION[SPC]
TAN[SPC]CARA"
150 PRINTTAB(2)"COMO[SPC]SEA[SPC]POS
IBLE."
170 PRINT"[2CRSRD]";TAB(4)"[RVSON][GRN]
PULSA[SPC]UNA[SPC]TECLA[RVSOFF][BLU]
"
180 GETA$:IFA$=""THEN180
190 FORN=1 TO 5
200 PRINT"[CLR][2CRSRD]"
210 PRINTTAB(2)"TE[SPC]QUEDAN[SPC]$:
";CASH
220 PRINTTAB(2)"EL[SPC]CUADRO[SPC]#"
;N;"ESTA"
230 PRINTTAB(2)"A[SPC]LA[SPC]VENTA."

240 PRINTTAB(2)"[CRSRD][RVSON][CYN]E
NTRA[SPC]TU[SPC]PUJA:[BLU]";
250 INPUTBID$
260 BID=VAL(BID$)
270 IFBID>CASHGOTO200
280 OP=INT(RND(1)*100)*10+150
290 PRINT"[2CRSRD]OTRO[SPC]COLECCION
ISTA"
300 PRINT"OFRECE[SPC]$";OP;"POR[SPC]
ESE"
310 PRINT"CUADRO.[CRSRD]"
320 IFBID>OPTHEN350
330 PRINTTAB(2)"LO[SPC]PERDISTE!"
340 GOTO390
350 PRINTTAB(2)"LO[SPC]HAS[SPC]COMPR
ADO!"
360 NU=NU+1
370 COLLECT(NU)=BID
380 CASH=CASH-BID
390 PRINT"[2CRSRD]";TAB(6)"[RVSON][GRN]
PULSA[SPC]UNA[SPC]TECLA[RVSOFF][BLU]
"
400 GETA$:IFA$=""THEN400
410 NEXT N
420 FORN=1 TO NU
430 PRINT"[CLR][2CRSRD]"
440 PRINTTAB(2)"PUEDES[SPC]VENDER[SPC]
AHORA"
450 PRINTTAB(2)"TU[SPC]CUADRO[SPC]#"
;N;"."
460 NB=INT(RND(1)*6)+1
470 FOR N2=1 TO NB
480 BID=INT(RND(1)*200)*10+500
490 PRINTTAB(2)"[2CRSRD]TE[SPC]OFREC
EN[SPC]$";BID
500 PRINTTAB(2)"POR[SPC]ESTE[SPC]CUA
DRO.[CRSRD]"
510 PRINTTAB(2)"TU[SPC]PAGASTE[SPC]$
";COLLECT(N);"."
520 PRINTTAB(2)"ACEPTAS[SPC](S/N)?"
540 GETA$:IFA$=""THEN540
550 PRINT"[CLR][2CRSRD]"
560 IFA$="S"THEN580
570 GOTO600
580 CASH=CASH+BID
590 GOTO690
600 NEXT N2
610 PRINT"[CLR][2CRSRD]"
620 PRINTTAB(2)"LO[SPC]SIENTO,[SPC]A
MIGO!"
630 PRINTTAB(2)"ESA[SPC]FUE[SPC]LA[SPC]
ULTIMA"
640 PRINTTAB(2)"OFERTA!.[SPC]TE[SPC]
HAS"
650 PRINTTAB(2)"PASADO.[2CRSRD]"
670 PRINTTAB(6)"[RVSON][GRN]PULSA[SPC]
UNA[SPC]TECLA[RVSOFF][BLU]"
680 GETA$:IFA$=""THEN680
690 NEXT N
700 PRINT"[CLR][2CRSRD]"
710 PRINTTAB(2)"COMENZASTE[SPC]CON[SPC]
$5000."
720 PRINTTAB(2)"TERMINASTE[SPC]EL[SPC]
JUEGO"
730 PRINTTAB(2)"CON[SPC]$";CASH;"."
740 PRINT"[2CRSRD]"
750 PRINTTAB(2)"JUEGAS[SPC]MAS[SPC](
S/N)?"
760 GETA$:IFA$=""THEN760
770 IFA$="S"THENRUN
```

EXCURSION POR LA MICROPROGRAMACION

Cursillo de Lenguaje Máquina —IV—

Por Diego ROMERO

En el último número vimos los modos de direccionamiento absoluto e inmediato, y en este vamos a ver el de página cero y relativo.

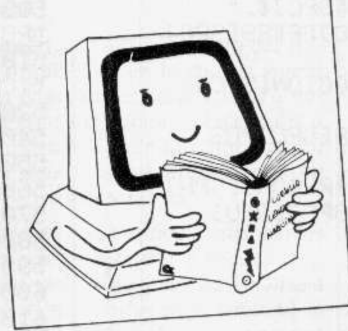
Direccionamiento en Página Cero

Este modo de direccionamiento aumenta la velocidad de ejecución de los programas en código máquina y ahorra memoria al acortar la longitud del programa.

Se trata de una versión recortada del modo de direccionamiento absoluto. En este último utilizábamos 3 bytes para cada instrucción (1 de instrucción y 2 de dirección de operando), ya que teníamos que indicar a cualquiera de las 65536 posiciones de memoria, pero cuando lo hacemos a las primeras 256 el byte de mayor peso de la dirección es siempre cero, y nuestras CPU's están dotadas de otro potente medio de direccionamiento que permite omitir el byte alto, con esto se ahorra uno de cada tres bytes y se incrementa la velocidad de ejecución al no tener que leer toda la dirección.

Antes veíamos que STA $ØØ33 era $8D $33 $ØØ, y con este modo dos bytes!... Repetimos que con STA $33 que es $85 $33... ¡sólo dos buyes!... Repetimos que con este modo de direccionamiento sólo podemos acceder a las 256 primeras direcciones, pero tiene sus ventajas...

El equivalente a esta instrucción en página cero sería POKE 51,A mientras que en absoluto sería POKE ØØ51,A. Podéis ver que son lo mismo pero con más números.



Direccionamiento Relativo

Este modo de direccionamiento saltar hacia adelante o detrás desde el punto donde se ejecuta el programa. Se salta cuando se cumple una condición determinada, y es equivalente a las instrucciones IF... THEN... o IF... GOTO... en BASIC, con la diferencia de que la condición que se comprueba es el estado de las banderas que hemos visto en el segundo capítulo de esta serie (número 4 de la revista), al estudiar el registro de estado de la CPU. Se puede saltar si el resultado de la última operación fue igual a cero, negativo, etc...

Lo complicado de este modo de direccionamiento puede ser entender el modo de especificar a dónde saltamos, ya que en BASIC decimos el número de línea al que saltamos, pero aquí se salta una "distancia" a partir del punto donde nos encontramos, veamos un ejemplo:

```
1Ø A=PEEK(254)
2Ø IF A=Ø GOTO 4Ø
3Ø POKE 255,A
4Ø END
```

Este programa leería lo que contiene la posición 254 y si es un cero termina, si no es cero lo pone en la posición 255. El equivalente en código máquina sería:

```
LDA  $FE   $A5  $FE
BEQ  $Ø2   $FØ  $Ø2
STA  $FF   $85  $FF
RTS        $6Ø
```

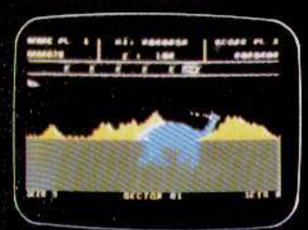
Lo primero que debemos aclarar es la nueva instrucción que aparece, se trata de ReTurn from Subrutine (vuelve de subrutina), se usa para volver el control al programa principal al igual que en BASIC la instrucción RETURN.

Lo segundo es el salto BEQ $Ø2, se trata de la instrucción Branch if EQual (salta si es igual a cero), y su operando el $Ø2 es "dos bytes" hacia delante a partir de la siguiente instrucción, si fuese BEQ $ØØ sería igual que no ponerla ya que continuaría en la siguiente instrucción. Si queremos que el salto sea hacia atrás, debemos ponerlo en complemento a dos, por tanto el salto sea hacia atrás, debemos ponerlo en complemento a dos, por tanto el salto puede estar en el rango de +128 a -127 posiciones de memoria. Esto ya lo iremos viendo más claramente con ejemplos en código máquina ya que en BASIC no tiene equivalente como he dicho antes. ■

indescomp SOFTWARE

UNA NUEVA GENERACION DE VIDEOJUEGOS

COMMODORE 64



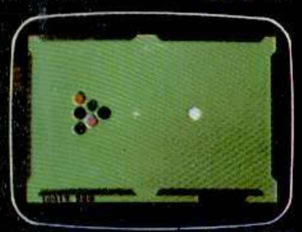
MUTANT CAMELS — Cass. 2 200



MATRIX — Cass. 2 000



CHOPFLITER — Cart. 8 170



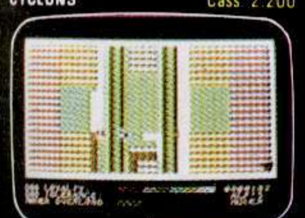
HUSTLER — Cass. 2 200



REVENGE MUTANT CAMELS — Cas. 2 200

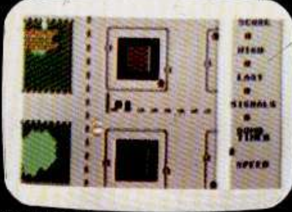


CYCLONS — Cass. 2 200



HOVER BOVER — Cass. 2 200

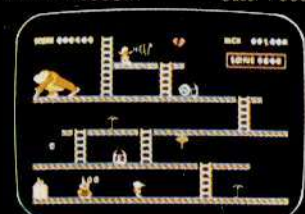
COMMODORE 64



SIREN CITY — Cass. 1 900



CUDDLY CUBURT — Cass. 1 900



KONG — Cass. 2.000



SKRAMBLE — Cass. 1.700



MOON BUGGY — Cass. 2 000

COMMODORE 64

• SPRITE MAN	1 900
• FROGGER 64	1 700
• SNAKE	1 800
• ANIHILATOR	1 600
• PAKACUDA	1 700
• EXTERMINATOR	1 900
• VORTEX RAIDER	1 700
• DEFENDER 64	1 700
• LASER ZONE	1 700
• CHOPFLITER	8 170
• ZAXXON	7 190
• JUMPMAN	7 190
• FORT APOCALIPSE	6 290
• FUTBOLIN	1 800
• HORACIO	2 200
• PROCESADOR TEXTOS	3 300
• MAILING Y ETIQUETAS	3 300
• CONTABILIDAD PERSONAL	3 300
• BASE DE DATOS FICHERO	3 300
• ULTRABASIC	2 900
• SINTHY 64	2 900
• FORTH TINY	2 900

SINCLAIR SPECTRUM*

- PENETRATOR
- TIME GATE
- EL JUGADOR DE AJEDREZ
- ASTROBLASTER
- COMECOCOS
- HORMIGAS
- TRAXX
- DISEÑADOR DE JUEGOS
- LA PULGA (BUGABOO)
- BILLAR AMERICANO
- EL PINTOR
- MISILES
- RAPTORES DE LA GALAXIA
- WINGED AVENGER
- INVASORES & GALAXIANS
- FORTH
- BASE DE DATOS
- MATEMATICAS
- ARITMETICAS BASICAS
- TUTOR

APPLE 900 PROG

ATARI 800 PROG

SINCLAIR ZX81 100 PROGRAMAS

DRAGON 32 60 PROGRAMAS

VIC 20 200 PROG

IBM 350 PROG

SOLICITE GRATIS CATALOGO GENERAL INDICANDO EL ORDENADOR SOBRE EL QUE LO DESEA

BUGABOO LA PULGA — Cassette 2 000

Ya a la venta en



El Corte Inglés — Sonitel

CHIPS & TIPS

Distribuidores autorizados y por Correo

ENVIAR A: indescomp, Castellana, 179. Tel. 6563012. Madrid-16

CANTIDAD	PRODUCTO	PRECIO UNIT.	TOTAL
EMPAQUETADO Y TRANSPORTE	GASTOS DE ENVIO	300	
		TOTAL	

Incluyo cheque nominativo a favor de INDESCOPM, S.A. por .... Pts.

Remitan el pedido contra reembolso a:

D. .... Dirección ....

n.º .... Provincia .... Tel. .... Profesión ....

* Los programas para Spectrum son comercializados únicamente y en exclusiva por Investronica, S. A.

Estamos en la Feria de la Informática.
"EL CORTE INGLES" Castellana, del 4 al 28 de Abril.

CUANDO SE TIENE UN COMMODORE 64 ES MUY DIFICIL SER MODESTO

Cuando se tiene un ordenador personal con 64K de memoria, una magnífica resolución, 16 colores, efectos tridimensionales con "sprites", un sonido equivalente al de un sintetizador, un teclado profesional con 62 caracteres gráficos, toda una amplia gama de periféricos profesionales, la más completa serie de programas educativos, profesionales y de video-juegos...; en resumen, cuando se tiene un ordenador personal como no existe ningún otro en el mercado y el más vendido mundialmente, cuando se tiene el Commodore 64, es muy difícil mostrarlo sin que el orgullo se te note.



EL ORDENADOR PERSONAL DE LA FAMILIA MAS POTENTE

– Sistemas de gestión profesionales series 8000 Y 700. – Ordenador portátil SX 64.
– Ordenador personal COMMODORE 64. – Ordenador familiar VIC 20.



MICROELECTRONICA Y CONTROL, S.A.
c/ Taquígrafo Serra, 7, 5.° BARCELONA-29 c/ Princesa, 47, 3.° G MADRID-8

SLOGAN